

Intro to HTML5, CSS3, and Javascript

Richard Clark, Kaazing





Part 1: How did we get here?

A Moment of History

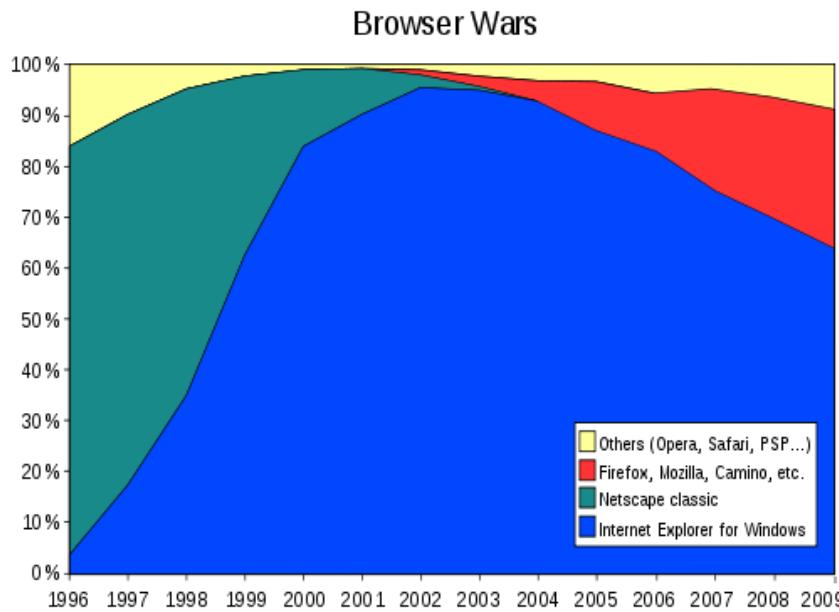


Web Images Groups News Froogle LocalNew! more »
[Advanced Search](#) [I'm Feeling Lucky](#)

[Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2006 Google

2006



- 1-2 dominant browsers
- Static pages
 - ✓ Link-heavy
 - ✓ Server-rendered
- Dependent on plug-ins
- W3C Mobile Web Seminar
 - “Who should attend?
 - *Authors, pioneers, visionaries, web designers, enthusiasts, freelancers, project managers, and developers...*”

Image source: [Wikipedia](#)

Today

Make Y! your homepage

YAHOO!

Tuesday, August 21, 2012

SIGN UP TODAY

FANTASY FOOTBALL

It's game time

- Create a league or join one, for free

YAHOO! SITES

- Autos
- Dating
- Finance (Dow)
- Flickr
- Games
- Horoscopes
- Jobs
- Mail
- Messenger
- Movies
- Music
- My Yahoo!
- News
- omg!
- Real Estate
- Screen
- Shine
- Shopping
- Sports
- Travel
- TV
- Weather (77°F)

More Y! Sites

FAVORITES

Web Images Video Local News More

Search

TRENDING NOW

- 01 Dick Van Dyke
- 02 Kelly Ripa co-host
- 03 Robin Roberts
- 04 iPhone 5 release date
- 05 Phyllis Diller
- 06 Rosie O'Donnell he...
- 07 Lisa Whelchel Surv...
- 08 BP
- 09 U.S. Army Nazis
- 10 Psoriasis drugs

Watch the show >

U.S. Cellular

Goodbye asterisks. Hello straightforward.

Click for Confessions

Visit us on YouTube - Ad Feedback

MUST-SEE VIDEOS ON YAHOO!

Firefighter stays on job despite major loss

+Richard Search Images Maps Play YouTube News Gmail Documents C

Enter stock symbol Get Quotes

LIFESTYLES

'Fifty Shades'... of green

One of America's largest bookellers gets good news -- thanks in part to the book

What 46% of women lie to their partners about

Change background image

Google Search I'm Feeling Lucky

Advertising Programs Business Solutions Privacy & Terms +Google About Google

SET EDITION: U.S. INTERNATIONAL MÉXICO ARABIC

TV: CNN CNN en Español HLN

Home TV & Video NewsPulse U.S. World Politics Justice Entertainment Tech Health Living Travel Opinion iReport Money Sports

updated 6:08 PM EDT, Tue August 21, 2012

IN THE NEWS Electoral Map • Todd Akin • Mars • Diana Nyad • 'Honor' Killing • Phyllis Diller • Best Places to Live • Tony Scott • Inside Syria • Photos

Akin defies pressure to quit race before deadline

What made Scott's movies different

Director Tony Scott, who died in an apparent suicide Sunday, was known for action movies, but what made his films distinctive was a feel for the human side of things. FULL STORY | OPINION: TENSE FILMS FOR JITTERY TIMES | PROBE

FEATURED

Jellyfish, weather end Nyed's swim

Life lessons from two iconic women

New tech could bring 3D flicks, no glasses

America's top-earning towns

Politician by day, MMA fighter by night

'Ma' Barker shootout house for sale

Shows and Schedules

BONOBOS Men's Clothing

Fit + COLOR

Get 20% OFF*

*Valid for new customer only

FREE SHIPPING & RETURNS

ADVERTISING

ON LIVE TV NOW

The Situation Room with Wolf Blitzer Wolf Blitzer brings you the latest in political news and international events.

Watch LIVE TV

TV & Video »

ON LIVE TV NOW

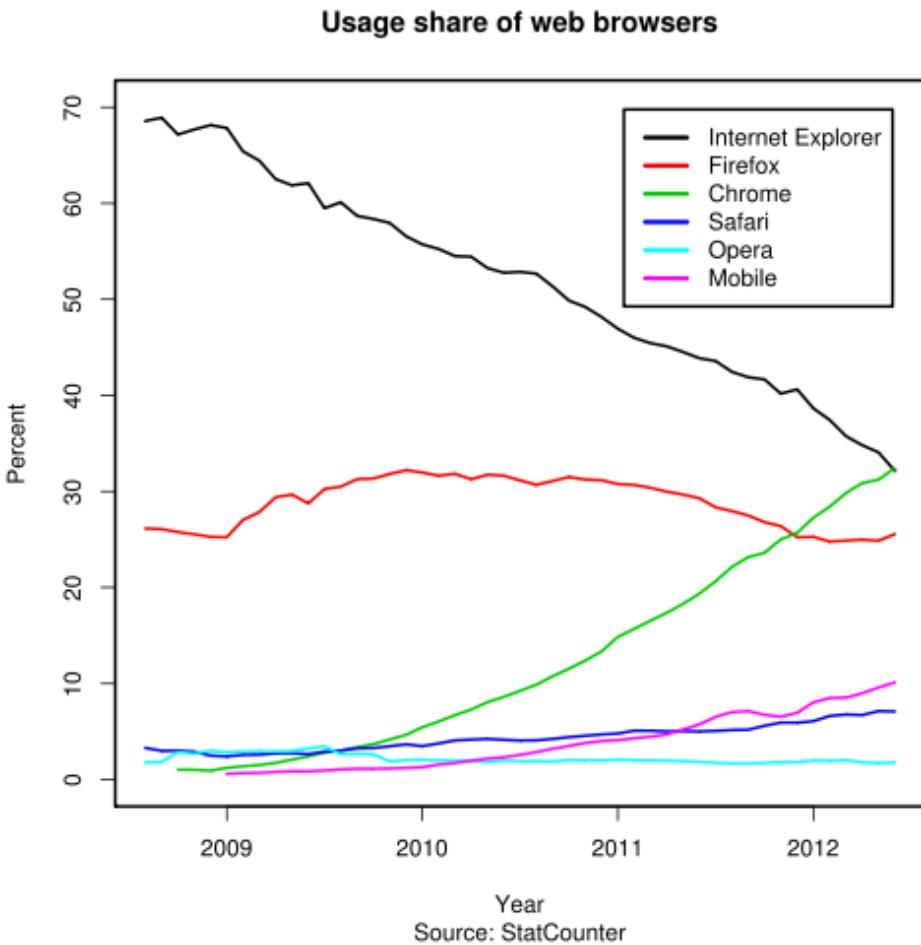
The Situation Room with Wolf Blitzer Wolf Blitzer brings you the latest in political news and international events.

Watch LIVE TV

FEATURED TV

Canceled his Plara Morgan interview

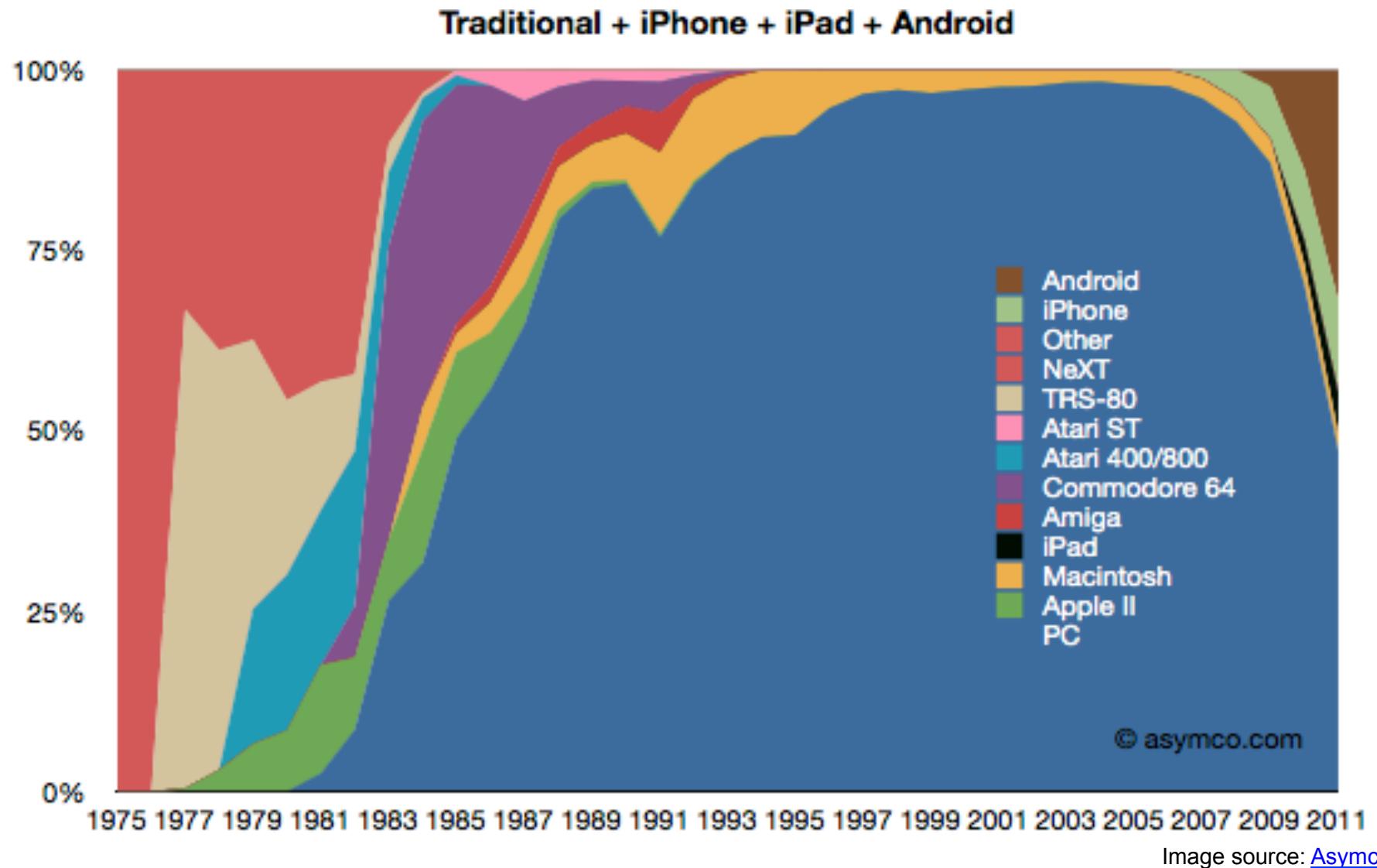
Todd Akin's empty chair



- Webkit, Mozilla, new IE
- Interactive pages
 - ✓ Audio / Video
 - ✓ Interactive features
 - ✓ Personalization
 - ✓ Applications (e.g. gmail)
 - ✓ Mash-ups, social, etc.
- And about mobile...

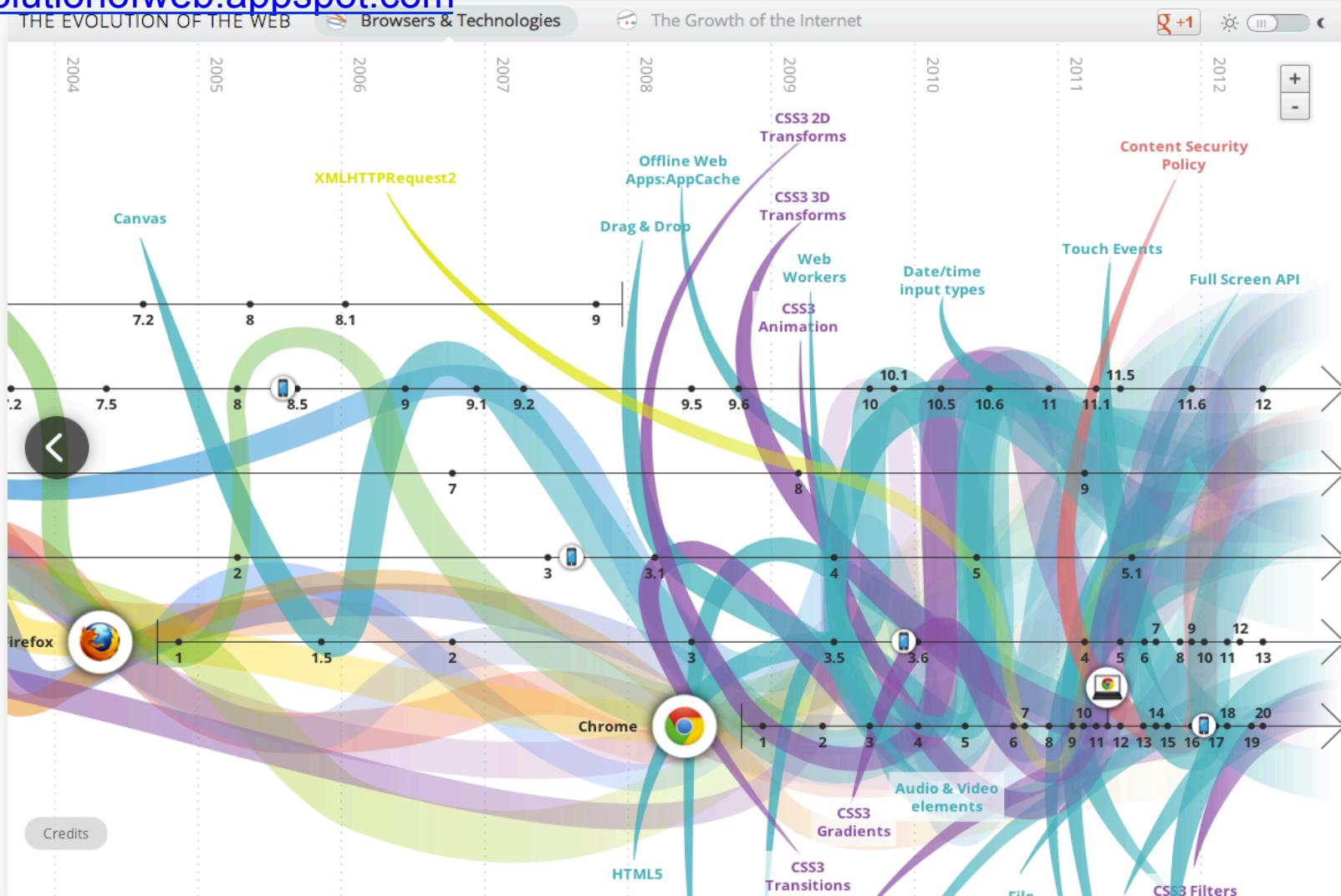
Image source: [Wikipedia](#)

Mobile Floods In



Browsers change rapidly

evolutionofweb.appspot.com



The origins of HTML5

- Who: Browser Manufacturers
- Why:
 - ✓ Create browser as app platform
 - ✓ Simplify complexity of web authoring



Timeline



What HTML5 Changes

- ✓ Greater separation of content and styling
- ✓ Greater emphasis on client-side interactivity
- ✓ Wider range of tools available (many open-source)
- ✓ Greater need for modern software engineering techniques

Key Skills for Modern Devs

- ✓ Separating content from style
- ✓ Writing clean “semantic” markup
- ✓ CSS layout techniques
- ✓ Adapting to browser differences
- ✓ Developing & debugging interactive code



Let's get started

Getting started

- Who's on your teaching team?
- What's our intention for today?
- Ground rules
- Logistics: Planned timing, materials, bathrooms, etc.
- Your turn:
 - What's your background?
 - What's important to you?

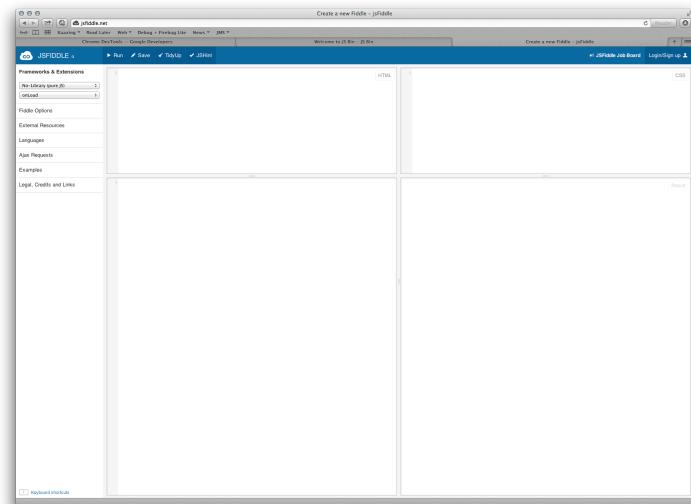


Making a fresh start

Tools for today (1): A text editor

- Either:

- A desktop editor
- JSFiddle: <http://jsfiddle.net>
- JSBin: <http://jsbin.com>
- Codio <http://codio.com>



Tools for today (2): Chrome

<https://developers.google.com/chrome-developer-tools/>

The screenshot shows the official Google Developers documentation for Chrome DevTools. The page features a navigation bar at the top with links like Home, Products, Conferences, Showcase, Live, Groups, and a search bar. A sidebar on the left contains a table of contents for the 'Authoring and Development Workflow' section, including topics such as Editing Styles and the DOM, Managing Application Storage, Evaluating Network Performance, Debugging JavaScript, and Performance Profiling with the Timeline. The main content area displays a large image of a laptop screen showing the Chrome DevTools interface with the 'Elements' panel open, inspecting an element from a web page. Below the image is a heading 'Overview' followed by a detailed description of what DevTools are and how they can help developers. A note at the bottom of this section suggests using Google Chrome Canary for the latest version. The bottom half of the page is a detailed view of the DevTools interface, showing the Elements tab selected, displaying the HTML code of a two-column document. The right side of the interface shows the Computed Style, Styles, and Matched CSS Rules panels, which are part of the Elements tool's functionality.

Tools for today (3): The HTML5 Specification

<http://whatwg.org/>

The screenshot shows the homepage of the WHATWG (Web Hypertext Application Technology Working Group) community. The page features a large green question mark icon at the top center. Below it, the text "Welcome to the WHATWG community" and "Maintaining and evolving HTML since 2004". The page is organized into a grid of nine colored boxes:

- HTML** (Yellow box): Read, use, or implement the HTML Living Standard.
- Web Dev Edition** (Teal box): An edition of the HTML standard designed for Web Developers.
- Join** (Pink box): Comment on the HTML standard and send proposals of your own.
- Wiki** (Green box): Poke at our wiki pages and see what you unearth.
- Help** (Red box): Send questions and help others in the help@whatwg.org mailing list.
- Forums** (Blue box): Talk with Web designers about how to write HTML.
- FAQ** (Purple box): Get answers to your questions about HTML and the WHATWG.
- Specs** (Orange box): See the other specifications developed at the WHATWG.
- IRC** (Pink box): Chat with other members of the WHATWG community.

At the bottom of the page, a note states: "Queries should be directed either to one of our [mailing list](#) or to [Ian Hickson](#), who is acting as a spokesman for the group."

Start with a simple page

```
<!DOCTYPE html>

<html lang="en">

<head>
    <meta charset=utf-8>
    <title>Greetings</title>
</head>

<body>
    <h1>Hello, world</h1>
</body>
</html>
```

Label the structural elements

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset=utf-8>
    <title>Greetings</title>
</head>
<body>
<b><header>A greeting page</header></b>
<b><main><h1>Hello, world</h1></main></b>
<b><footer>Copyright? We don't need no steenking
copyright</footer></b>
</body>
</html>
```

A few more basics

- Block elements: h1-h6, p, div, blockquote, pre, br, ...
- Inline elements: span, b, i, s, u, cite, ...
- Lists: ul, ol, dl; li, dt, dd
- Linking: a

Embedding content

- `img`
- `video`, `audio`
- `iframe`, ...
- Tables (next slide)
- Forms (later)

Tables

- `table`,
- `tr/td/th`
- `thead/tbody/tfooter`
- `colgroup/col`

Simple markup exercise: Modernizing Markup

- Get the text of Alice in Wonderland from the Gutenberg Project
 - Download from <http://www.gutenberg.org/files/11/11-h/>
- Create a new blank HTML5 page
- Copy Chapter 1 into it
 - Don't worry about styling
- Ask around for help if you need it!



Adding styles with CSS

How CSS Works

- CSS **Selectors** match parts of the HTML page
- CSS **Properties** change how that part is drawn
- Example
 - *p { font: 12px/14px sans-serif }*
 - *p* is the **selector**
 - *font: 12px/14px sans-serif* declares the **font properties**
 - *font* is actually a **shorthand property** that breaks down to:
 - o *font-style: normal*
 - o *font-size: 12px*
 - o *line-height: 14px*
 - o *font-family: sans-serif*

Core CSS Selectors

Selector Type	Example
Tag	body
ID	#part2
Class	.intro

body {
 font-family: Arial;
 font-size: 12pt;
 color: black;}
h1 { color: blue }
.intro {
 font-style: italic }

<b**ody**>
<b**1**>Hello, World</b1>
<p **class="intro"**>
This is my first page
</p>
<p **class="intro"**
id="part2">
And some extra content
here.
</p>
</b**ody**>

CSS Colors

sixrevisions.com/css/colors-webdesign/

Color	Notes	
#00ff00	red, green, blue (hexadecimal)	
#f00	→ #ff0000	
rgb(0, 0, 255)	red, green, blue, each 0-255	
rgba(0, 0, 255, 0.5)	red, green, blue, alpha (no IE < 9)	
hsl(320, 100%, 25%)	hue, saturation, lightness (no IE < 9)	
indigo	See color names	

Aside: Hexadecimal

- Easy (if you have 16 fingers)
- Base 16: 0123456789ABCDEF
- Examples
 - $\#10 = (1 * 16) + 0 = 16$
 - $\#7F = (7 * 16) + 15 = 127$
 - $\#80 = (8 * 16) + 0 = 128$
 - $\#FF = 255$

CSS Inheritance

```
<body>
  <section id="intro">
    <h1>Introduction</h1>
    <p>This is an
    introduction</p>
  </section>
  <section id="details">
    <h1>Details</h1>
    <p>
      The details go here.
    </p>
  </section>
</body>
```

```
body {
  color: #888;
  font-weight: bold;
}

#intro {
  color: blue
}

#intro p {
  font-style: italic
}

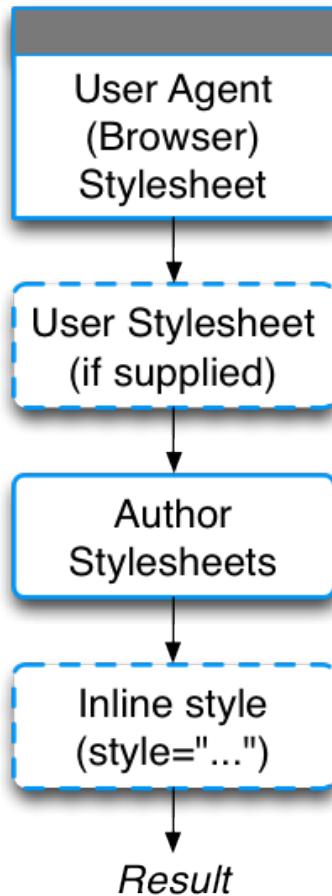
h1 {
  font-face: Verdana
}
```

Styling practice #1

- Using Alice in Wonderland
 - Copy the styles from the original into a stylesheet
 - o `<link rel="stylesheet" href="styles.css">` in the `<head>`
 - Change all of the paragraphs to red
 - Put a class on some of the tags, set that color to green
 - Put an id on one tag, set the color to blue
- Extra credit
 - What other styling can you change?
 - How would you override the styling on that id selector?

CSS: The Cascade

Style rules are combined via the [cascade](#)



```
h1 {  
    font: 18pt Times;  
    color: blue;  
}
```

```
h1 {  
    font-size: 12pt;  
}
```

Result:

```
font: 12pt Times;  
color: blue;
```

CSS: Conflicting Rules

Resolved via Specificity

```
<h1 id="first">Testing</h1>
```

1. Most specific rule wins

```
<p>This is a test</p>
```

2. Equal specificity? Later rule wins.

```
<h2>Testing what?</h2>
```

```
<p>Of the emergency  
broadcast system</h2>
```

```
<h1 class="more">Extra</h1>
```

weight	Matching selector	Example
1000	Local style attribute	
100	ID selector	#first
10	# of attributes and pseudo-classes	.more
1	# of element names & pseudo elements	h1

Specificity Examples

```
<h1 id="first">Testing</h1>
<p>This is a test</p>
<h2>Testing what?</h2>
<p>Of the emergency
broadcast system</h2>
```

```
<h1 class="more">Extra</h1>
```

```
<h1 style="color:indigo">
Levi's 501 Jeans
</h1>
```

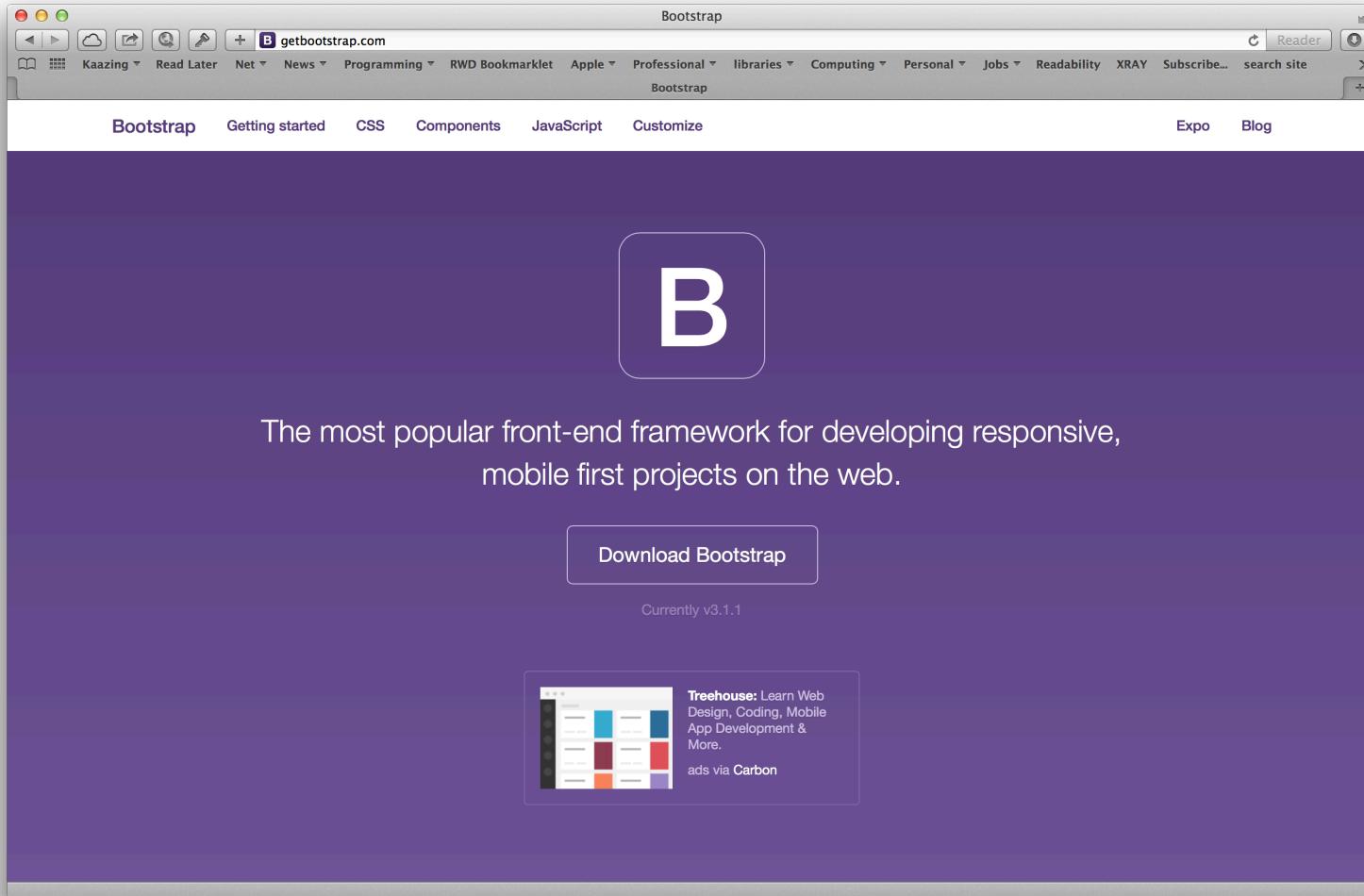
Selector	Score
h1	0010
#first	0100
h1 #first	0101
h1 .more	0011

Styling practice #2

- Using your existing page
 - What other styling can you change?
 - How would you override the styling on that id selector?



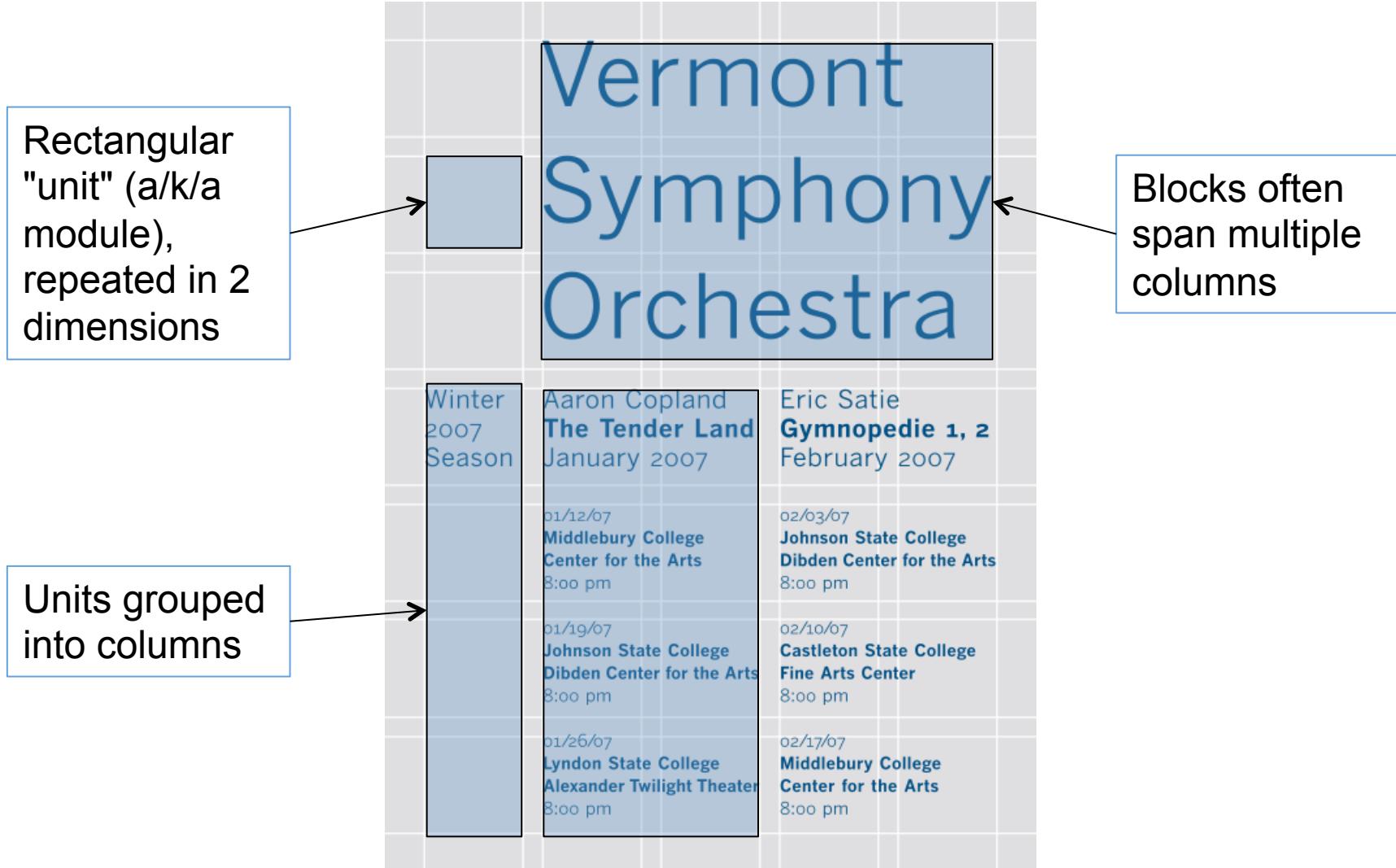
CSS: Twitter Bootstrap



Twitter Bootstrap

- Pre-made **CSS Framework**
- Simplifies layout with a **Grid System**
- Mostly works via CSS classes

Traditional grid systems



CSS Grids

Column-oriented

Uses class attributes to specify width.
Use block elements for vertical arrangement.

New (experimental)
[CSS Grid layout](#)
specification similar to
traditional 2D
typographical grid.

IE 10 only for now.

Options:

Fixed-width (e.g. 960 px)

Proportional (e.g.
50%)

Responsive (via media
queries)

Some systems allow nested grids.

CSS Grids in Bootstrap

`p class="col-lg-4"`

`p class="col-lg-10 col-md-10 col-xs-12"`

`div class=col-lg-4`

`div class=col-lg-8:`

`div class=col-lg-10`

`div class=col-lg-12`

`div class=col-lg-12`

Help designing with grids

<http://javascriptgrid.org/>

The screenshot shows the homepage of The JavaScript Grid. At the top center is the logo "The JAVASCRIPT GRID". Below it are two main sections: "Take one of THE PRESSETS" on the left and "Create YOUR OWN GRID" on the right. The "PRESSETS" section features four grid system options: "960 GRID SYSTEM" (12 columns), "960 GRID SYSTEM" (16 columns), "blueprint" (DRAG ME!), and "Less Framework 4" (DRAG ME!). The "YOUR OWN GRID" section has input fields for "12 columns", "60 px column width", and "20 px gap width", each with minus and plus buttons. Below these is a "YOUR SNIPPET:" button with a "DRAG ME!" label. At the bottom, there's a browser window showing the grid system applied to a page with "Less 4|10" content, and a note: "Just drag the snippets into your bookmarks bar, open your URL and click the bookmark." A cursor is shown dragging the "DRAG ME!" button from the blueprint snippet.

Take one of
THE PRESSETS

960 GRID SYSTEM 12

960 GRID SYSTEM 16

blueprint DRAG ME!

Less Framework 4 DRAG ME!

DRAG ME!

DRAG ME!

Create
YOUR OWN GRID

12 columns

60 px column width

20 px gap width

YOUR SNIPPET:

DRAG ME!

DRAG ME!

DRAG ME!

DRAG ME!

DRAG ME!

DRAG ME!

Just drag the snippets into your bookmarks bar, open your URL and click the bookmark.

Less 4|10

Bootstrap Lab

- Change your page to include Twitter Bootstrap
- Drop the old stylesheet
- Add a table of contents to the page
 - Put it in a <nav> element
 - Contents Chapter 1Down the rabbit hole
- Move the TOC to the left of the content
 - Use column classes
- Extra challenge
 - Install the Responsive Web Design bookmarklet
<http://responsive.victorcoulon.fr>
 - Add column classes to work at different screen sizes



Laying out pages without Bootstrap

CSS Box Model

```
body {  
    background-color: gray;  
}  
  
h1 {  
    font: 18pt Garamond;  
    border: 2px solid blue;  
    background-color: white;  
    padding: 18pt;  
    margin: 16pt;  
}
```



The **display** property

- Block vs. inline layout (and other values)

CSS Positioning

www.alistapart.com/articles/css-positioning-101/

Position	Block position
static	@ normal place in the flow
relative	Offset from its position in the flow. <i>Establishes new coordinate system.</i>
absolute	(top, left, bottom, right) in enclosing coordinate system. <i>Removed from the flow.</i>
fixed	(top, left, bottom, right) in browser window. <i>Removed from the flow.</i>

Float and Clear

www.alistapart.com/articles/css-floats-101/

Float	Block position
left	Left edge, content flows around right; stackable
right	Right edge, content flows around left; stackable
inherit	Use parent's float value
none	In the normal flow

Clear	Block position
left	Below any float:left element; back in normal flow
right	Below any float:right element; back in normal flow
both	Below any floated element; back in normal flow
inherit	Use parent's clear value
none	Stay in the normal flow



Responsive Design in a Nutshell

Media Queries

```
/* Media-specific sections of stylesheet */

@media all and (orientation:landscape) {
    /* e.g. rotated smartphone */
}

@media screen and (max-device-width: 480px) {
    /* small screen */
}

@media print {
    /* drop navigation elements that make no sense on paper */
    nav { visibility:hidden; }
}
```

Media Types

www.w3.org/TR/2009/CR-CSS2-20090908/media.html

Media type	Notes
all	Any media
braille, embossed	Handheld reader or printed Braille
handheld	Handheld (mobile) device
print, projected	Paged media
screen	Bitmapped
speech	Screen reader
tty	Character grid
tv	“Low resolution, sound available”

Media Queries

Units can be pt, px, em, ex, dpi, dpcm, ...

Accepts min- and max- prefixes



“media feature”	Example	
width, height	@media screen and (min-width: 400px) and (max-width: 700px) { ... }	✓
device-width, device-height	@media screen and (device-width: 800px) { ... }	✓
orientation	@media all and (orientation:portrait) { ... } @media all and (orientation:landscape) { ... }	
aspect-ratio device-aspect-ratio	@media screen and (device-aspect-ratio: 16/9) { ... } @media screen and (device-aspect-ratio: 32/18) { ... }	✓
color color-index	@media all and (color) { ... } @media all and (min-color: 2) { ... }	✓
monochrome	@media all and (monochrome) { ... }	✓
resolution	@media print and (min-resolution: 300dpi) { ... }	✓
scan	@media tv and (scan: progressive) { ... }	
grid	@media handheld and (grid) { ... }	
-webkit-device-pixel-ratio	@media handheld and (-webkit-device-pixel-ratio: 2)	

Media Queries Mini-Lab (optional)

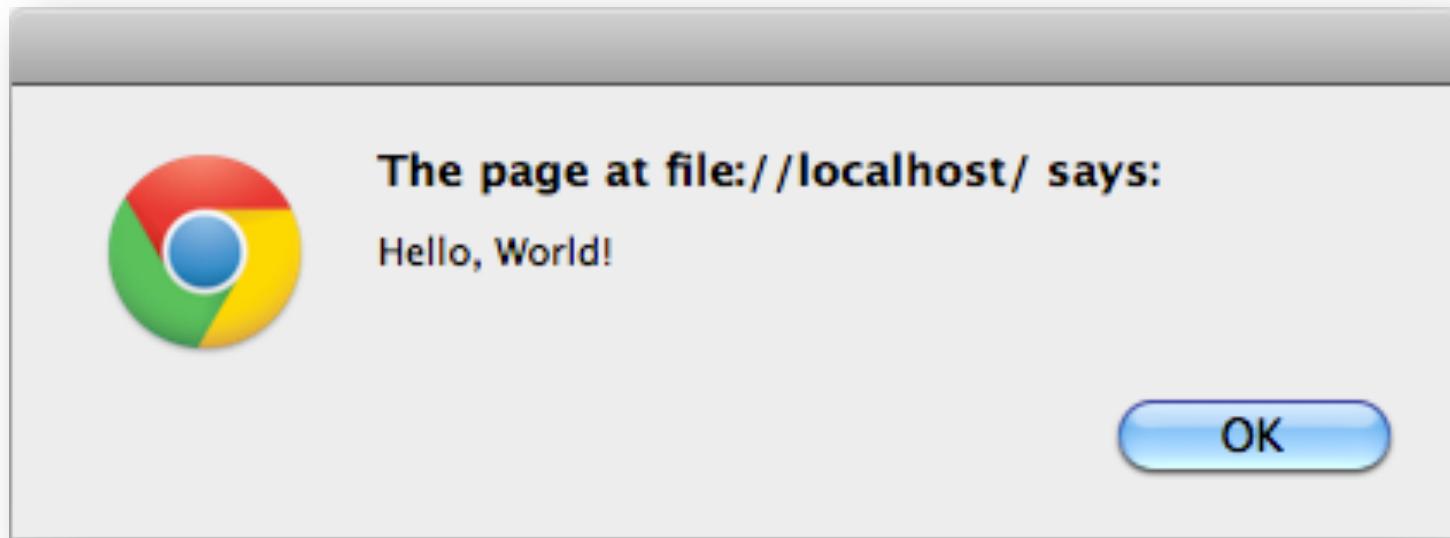
- Create a simple HTML5 page. *Do not use Bootstrap*
- Add some content (e.g. text from Alice)
- Define a `<style>` section with default styles
 - e.g. `p { font: 14px sans-serif }`
- Add media queries that change style for each screen size
 - e.g. `@media screen and (min-width: 640px) { p { color: blue } }`
- Play with it!

- Alternate lab:
 - Use your Twitter Bootstrap-enabled page
 - Install the Responsive Web Design bookmarklet
<http://responsive.victorcoulon.fr>
 - Add column classes to work at different screen sizes



Browser Scripting With Javascript

Hello, World!



```
<script>window.alert("Hello, World!")</script>
```

Programming with window

DOM API: dvcs.w3.org/hg/domcore/raw-file/tip/Overview.html

window	
Key attributes:	document, name, location, history, status
Common methods:	find, close, stop, focus, blur
User prompt methods	alert, confirm, prompt, print, showModalDialog

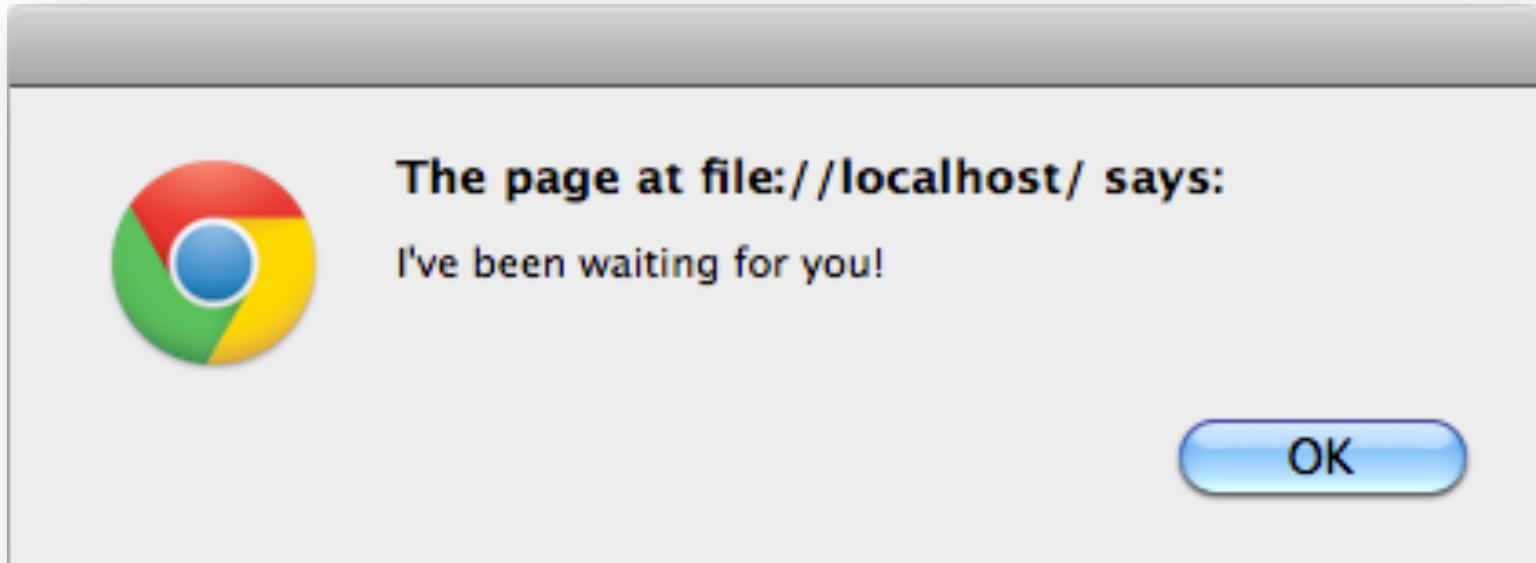
Practice:

- Add a <script> with `window.alert("Hello!")`
- Change "Hello!" to `window.name`
- Try
`window.status = "Hello!"`

A little more Javascript

- `window.alert("Hello!")`
- `var message = "Hello!";
window.alert(message);`
- `function makeMessage() {
 return "Hello!";
}
window.alert(makeMessage())`
- `var villain= "Carmen Sandiego";
function makeTitle(name) {
 return "Where in the World is " + name;
}
window.alert(makeTitle(villain));`

Let's wait for something to happen (events and functions)



```
window.onload = function() {  
    alert("I've been waiting for you!");  
}
```

The DOM environment

DOM API: dvcs.w3.org/hg/domcore/raw-file/tip/Overview.html

	window	document
Key attributes:	document, name, location, history, status	URL, documentURI, charset, doctype
Common methods:	find, close, stop, focus, blur	getElementsByName, getElementsByTagName, getElementsByClassName, getElementById, createElement, createDocumentFragment
User prompt methods	alert, confirm, prompt, print, showModalDialog	
Messaging	postMessage	
	<i>And many event handlers...</i>	

Event summary

en.wikipedia.org/wiki/DOM_events

body	image	Keyboard*	window
load, unload	abort	keypress	online ^t , offline ^t
		keyup, keydown	message
Mouse*			resize
click, dblclick		Forms	storage ^t
mousedown, mouseup		change	page hide/show
mousemove, mouseover, mouseout		select	load, unload
		reset	before/after print
			<i>App Cache^t</i>
General*		submit	<i>Drag & Drop^t</i>
blur, focus, error, load, scroll		<i>Key, mouse, general events</i>	

Applies to display elements (not body, frameset, br, script, title)

^t Explained later

Event handlers

```
function sayHello() { alert("Hello!") };  
var a = window.getElementById("a");
```

Type	Example	Notes
Inline		
Traditional	a.onclick = sayHello;	Only one
DOM API	a.addEventListener('click', sayHello, false); a.removeEventListener('click', sayHello, false);	Multiple handlers allowed

Non-standard IE < 8:

Type	Example	Notes
DOM API	a.attachEvent('click', sayHello, false); a.detachEvent('click', sayHello, false);	Multiple handlers allowed

Script examples

Go to a new page:

```
window.location = "http://www.kaazing.com";
```

Display alert:

```
window.alert("Hello, world!");
```

Respond to mouse clicks:

```
window.onload = function() {  
    var h1 = document.getElementsByTagName('h1')[0];  
    h1.onclick = function() {  
        h1.innerHTML = "Hello, World";  
    }  
}
```

Mini Lab

1. Build the *Hello, World* page (`window.alert`).
2. Build the `window.onload` page.
3. Challenge:
 1. Add a `<button>` to the page
 2. Add an event handler for when the button is clicked (`window.alert`)

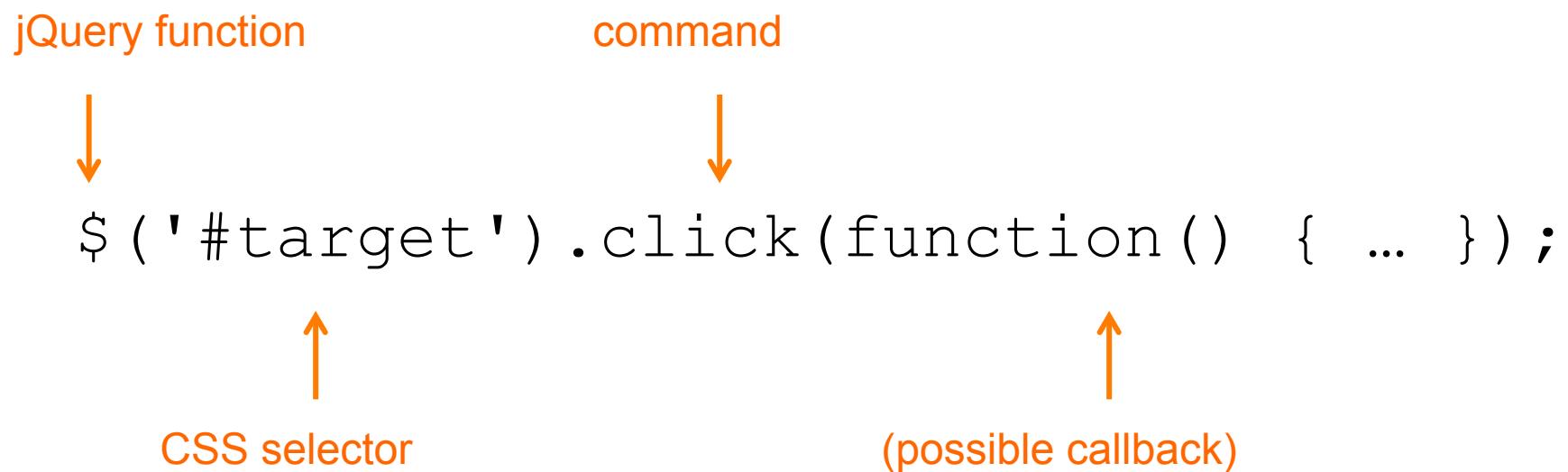
How do we do all this...

- Interactions
 - Click
 - Mouseover
 - Data entry (text, numbers, dates, ...)
- DOM changes
 - Add / remove elements
 - Show / hide
 - Alter styles
- Cache data locally

Meet JQuery

```
<!doctype html>
<html>
<head>
<meta charset=utf-8>
<script
  src="http://code.jquery.com/jquery-1.6.4.min.js">
</script>
<span id="target" style="border:1px solid">
  Click me!
</span>
<script>
$('#target').click(function() { alert('Hello, world') ; });
</script>
</html>
```

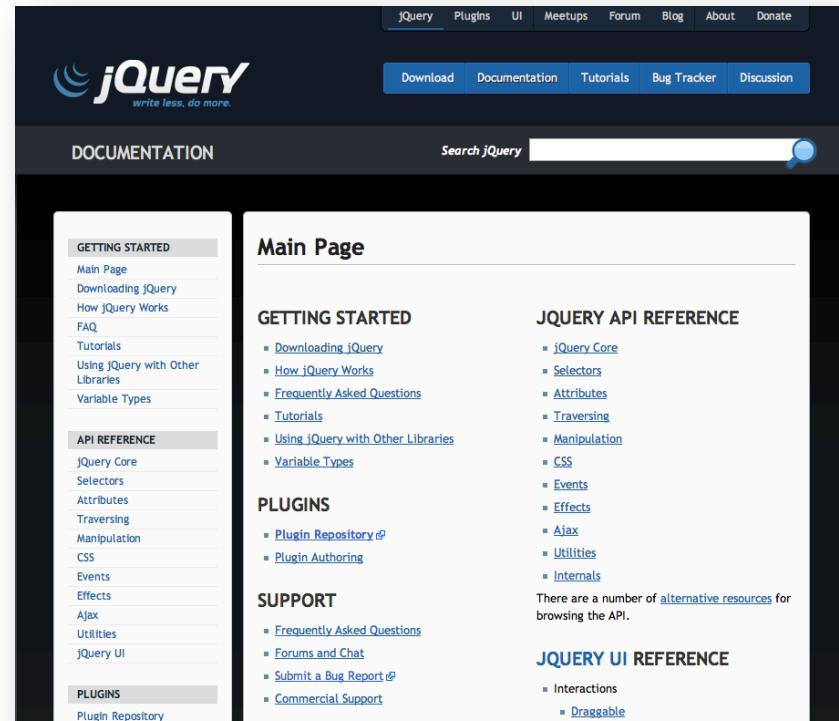
Using jQuery



Feature Hunt...

api.jquery.com

- Find the command to:
 - Add a CSS class
 - Get an attribute's value
 - Fade out an element
 - Append to a list
 - Attach data to a node



jQuery walk-through

- In JSFiddle:
 - Create a page with some content
 - Find an element (or elements) and fadeOut()
 - Apply styling instead
 - Build interactivity
 - Add a button
 - Add a click handler
 - Show a message
 - Add content to the page

jQuery in Action

demo.kaazing.com/portfolio

KAAZING®

Real-Time Portfolio via JMS and WebSockets

Company	Ticker	Price	Change	Shares	Value	Buy	Sell
3m Co	MMM	.34	▼ -0.01	424	144.16	<button>Buy</button>	<button>Sell</button>
AT&T Inc.	T	.25	▲ 0.01	18	4.50	<button>Buy</button>	<button>Sell</button>
Boeing Co.	BA	6.06	▼ -0.01	90	545.40	<button>Buy</button>	<button>Sell</button>
Citigroup, Inc.	C	5.73	▲ 0.07	6	34.38	<button>Buy</button>	<button>Sell</button>
Hewlett-Packard Co.	HPQ	37.98	▲ 0.54	0	0.00	<button>Buy</button>	<button>Sell</button>
Intel Corporation	INTC	.36	▲ 0.01	0	0.00	<button>Buy</button>	<button>Sell</button>
International Business Machines	IBM	56.28	▼ -0.29	22	1238.16	<button>Buy</button>	<button>Sell</button>
McDonald's Corporation	MCD	8.07	▲ 0.13	9	72.63	<button>Buy</button>	<button>Sell</button>
Microsoft Corporation	MSFT	54.11	▼ -0.93	0	0.00	<button>Buy</button>	<button>Sell</button>
Verizon Communications	VZ	1.15	▼ -0.02	0	0.00	<button>Buy</button>	<button>Sell</button>
Wal-Mart Stores, Inc.	WMT	.96	▼ -0.01	0	0.00	<button>Buy</button>	<button>Sell</button>
Total value:		569			\$2039.23		
Cash available:					\$2146.20		



Designing Applications

Use Cases

- Actors
- Systems
- Actions

Storyboards

Mockups

- Paper
- On-screen
- On-device

Exercise: Paper Prototyping

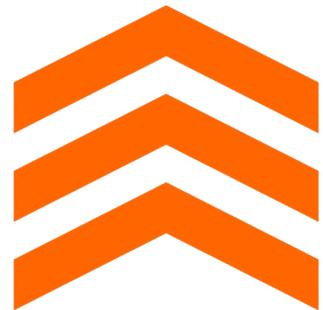
1. Select an app idea
2. Select one task (workflow)
3. Build mockups on 3x5 cards (a few screens)
4. Test with each other (be nice!)





A catalog of HTML5 features

HTML5 Feature Areas



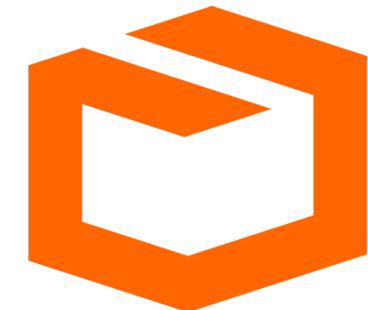
Semantics



CSS3



Multimedia



Graphics & 3D



Device Access



Performance



Offline & Storage



Connectivity

New Layout Tags

Layout (Semantics)

article	section
aside	nav
figure	figcaption
header	footer
details	summary
wbr	hgroup
time	mark
meter	ruby, rp, rt

Redefined (added semantics)

b, i, s, u

small, strong

menu (revived from HTML 4)

hr

Removed

Recommended alternatives for new development.

Removed	
tt, big, center, strike	Use CSS
font, basefont	Use CSS
frame, frameset, noframes	Use <code>iframe</code>
acronym	
applet	
dir	Changed to attribute

SEO via Microdata

- Emphasis on machine readability (SEO)
- Easy-to-write syntax (add attributes to any element)
- Uses microdata vocabularies @ data-vocabulary.org



CSS enhancements

- Focus on client-side dynamic styling
 - e.g. alternating table row colors
- New visual effects
 - Rounded corners, shadows, gradients, transforms

CSS3 Pseudo-Class Additions (1/2)

Selector	CSS	HTML
:target	p:target	 if URI ends in #foo
:enabled :disabled	input:disabled {color="grey"}	
:checked :indeterminate	input:checked {...} input:indeterminate {...}	<input type="radio" ...>
:root	:root	<html>...
:empty	p:empty	<p></p>
:not	i:not(:first)	<p><i>Hello</i>, </p><i>World</i>

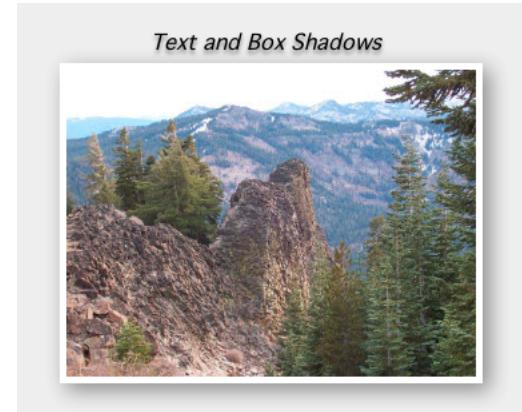
CSS3 Pseudo-Class Additions (2/2)

Selector	CSS	HTML
:first-child	div:first-child	<pre><div> <p>Hello</p> <p>World</p> </div></pre>
:last-child		
:first-of-type	div p:first-of-type	<pre><div> <p>Hello</p> </div></pre>
:last-of-type		
:only-child	div:only-child	<pre><div> <p>Hello</p> </div></pre>
:only-of-type		
:nth-child()	ol > li:nth-child(1) { color: red }	<pre>ol > li:nth-child(2n) {color:green} ol > li:nth-child(2n-1) {color:red} ol > li:nth-child(:odd) {color:red} ol > li:nth-child(:even) {color:red} ol > li:nth-child(:first) {color:red} ol > li:nth-child(:last) {color:red}</pre>
:nth-last-child()		
:nth-of-type()		
:nth-last-of-type()		

Visual effects



Making the Web Beautiful!

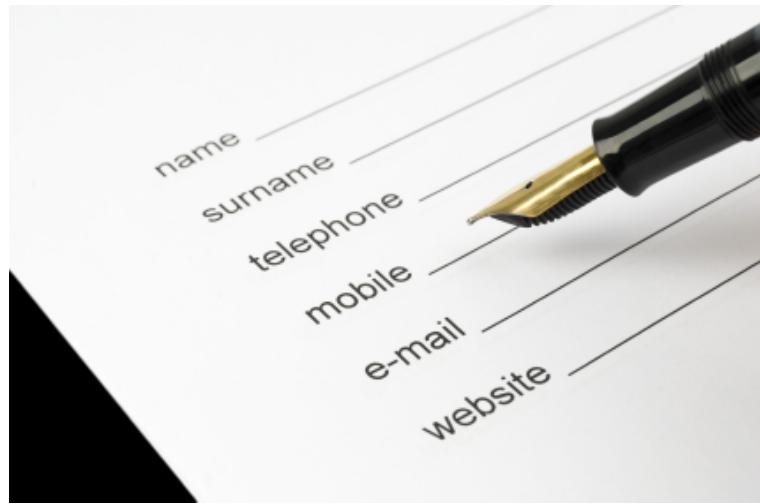


Guidelines

- Separate content, styling, and execution
- Use readable, semantic markup
- Reduce server-side device sniffing
 - Use media queries & Javascript to detect device characteristics
 - Available features
 - Screen size, orientation, resolution, etc.
- Design layouts for each device class

HTML5 Forms

- New form functionality
 - No JavaScript required
 - Native date and color pickers
 - Search, e-mail, web address
 - Client side validation
- Mobile optimized
- Backward compatibility built-in



Forms features

Runner: @peterlubbers

Tel #: 1-877-KAAZING

E-mail: peter Please enter a valid email address

DOB: Please enter a valid date

T-shirt Size:

- Small
- Medium
- Large

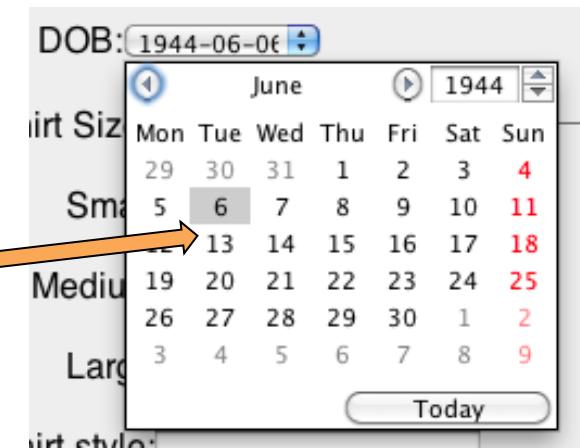
Shirt style:

Shirt Color:

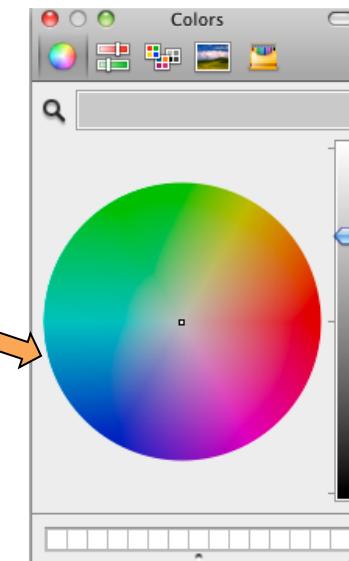
Expectations:

Confidence: 0%

Notes:



Date picker



Color picker

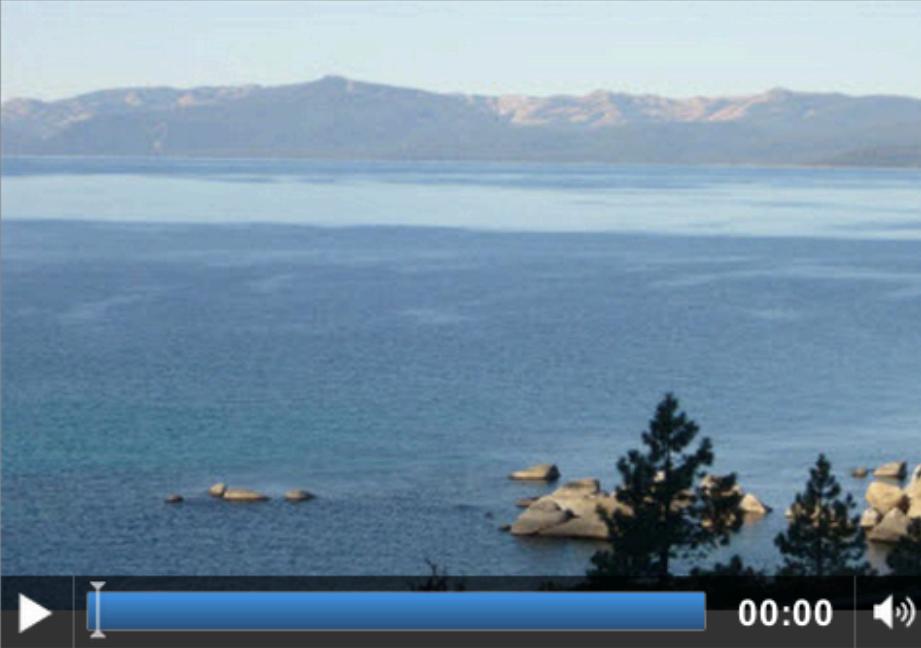
Native Audio/Video support

HTML5 Video

Video by Rocky

Links

YouTube



About

This video was created by Rocky Lubbers for the MediaFest 2010 Competition

Powered by HTML5

Audio/Video markup

```
// Video with multiple formats
<video>
  <source src="rocpoc.ogv" type="video/ogg">
  <source src="rocpoc.mp4" type="video/mpeg">
    A movie by Rocky Lubbers
</video>

// Simple markup with options
<video controls preload autoplay>
  <source src="rocpoc.mp4" type="video/mpeg">
    A movie by Rocky Lubbers
</video>
```

Codec Support

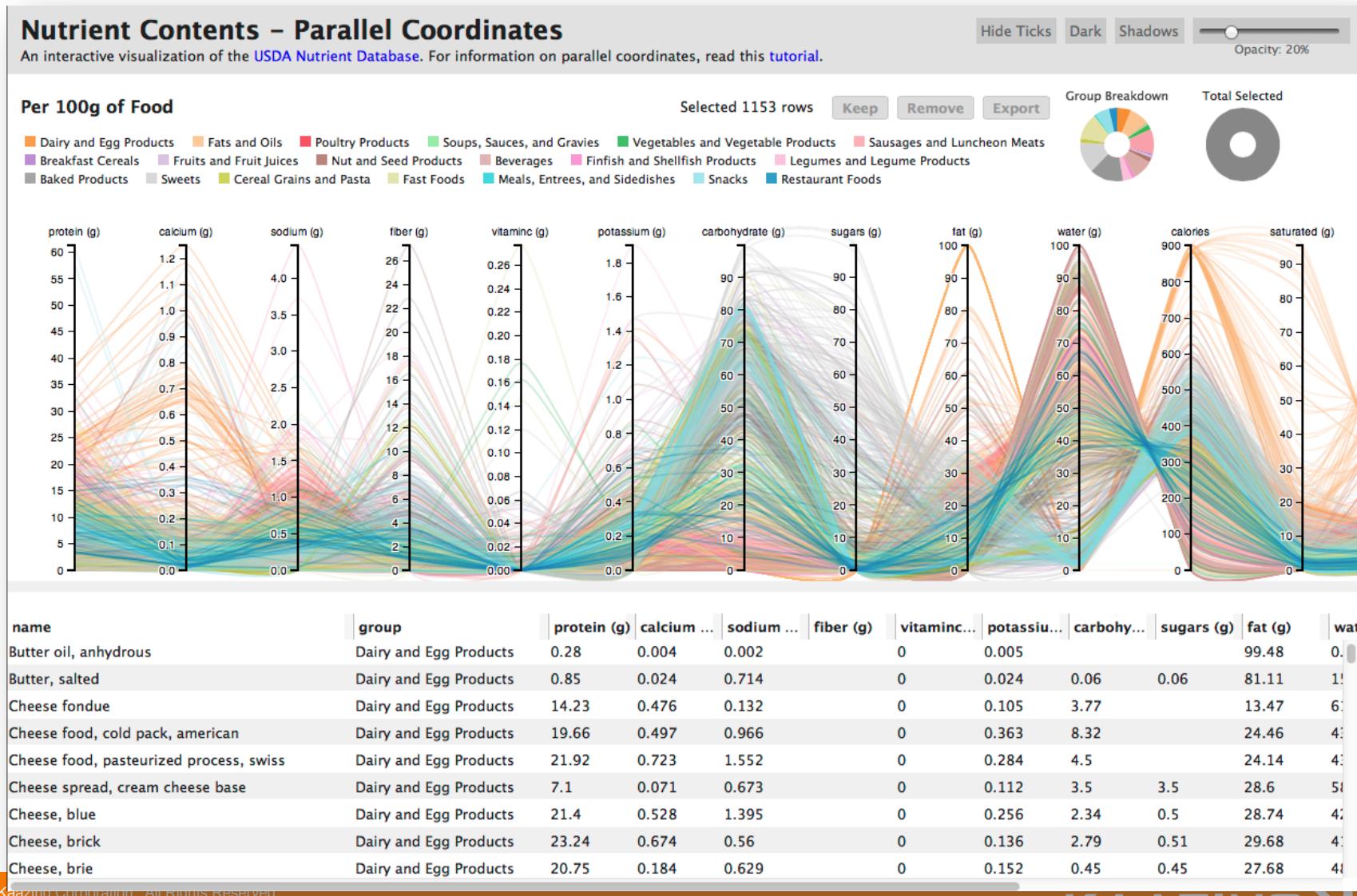
Browser	h.264	Ogg/WebM
Chrome	No [†] Yes	Yes
Safari	Yes	No
Firefox	No (Yes, in future)	Yes
Opera	No	Yes
Internet Explorer >= 9	Yes	No*

* Unless codec installed as a plugin

† Google said h.264 would be removed from Chrome, but never did it.

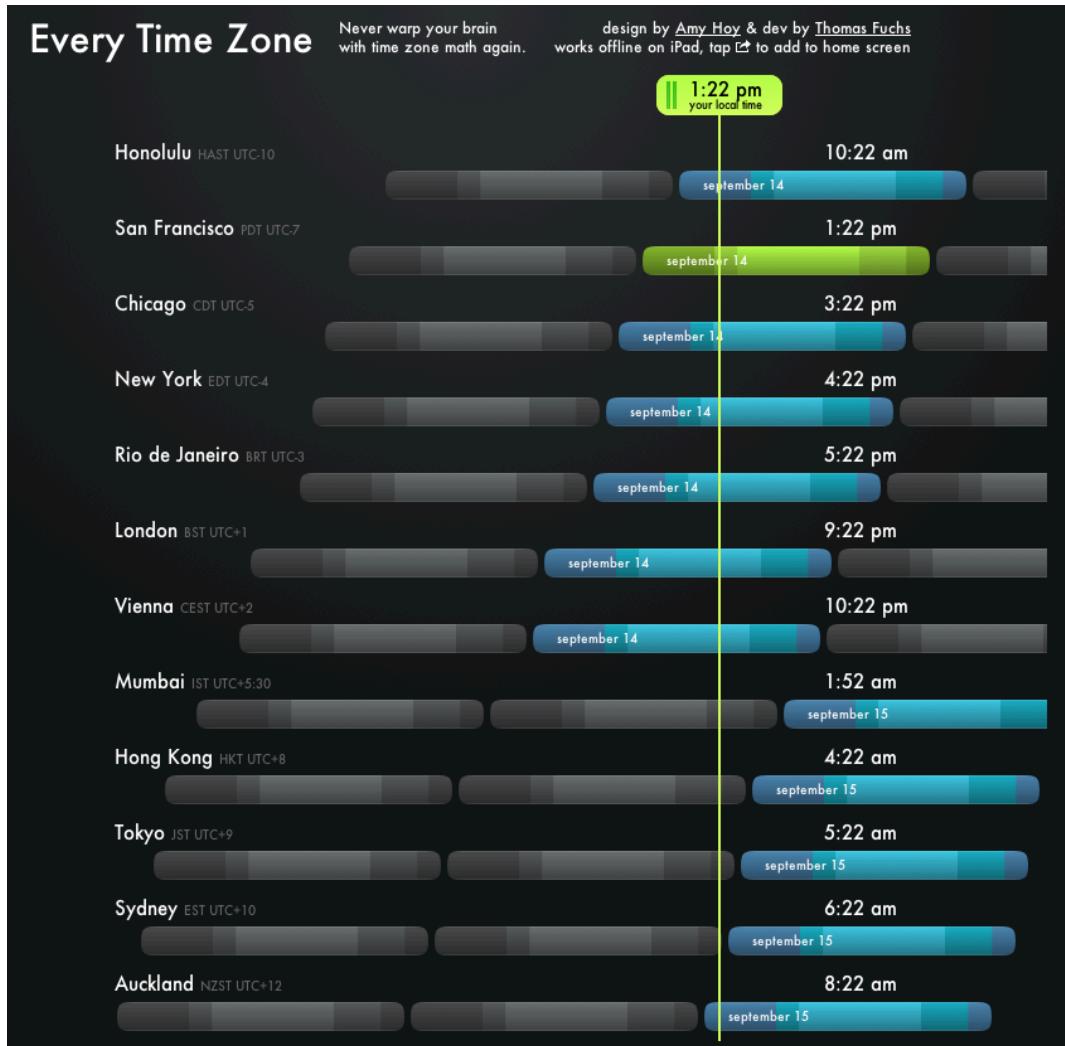
Interactive Graphics: SVG

exposedata.com/parallel/ (using [d3.js](#))



Interactive graphics: Canvas

everytimezone.com



Device Access

- Geolocation
- Accelerometer
- Camera, microphone, etc. (forthcoming)



Map tiles by [Stamen Design](#), under [CC BY 3.0](#).
Data by [OpenStreetMap](#), under [CC BY SA](#).

CORS

Cross-
Document
Messaging

Web
Workers

Web
Storage

Offline
mode

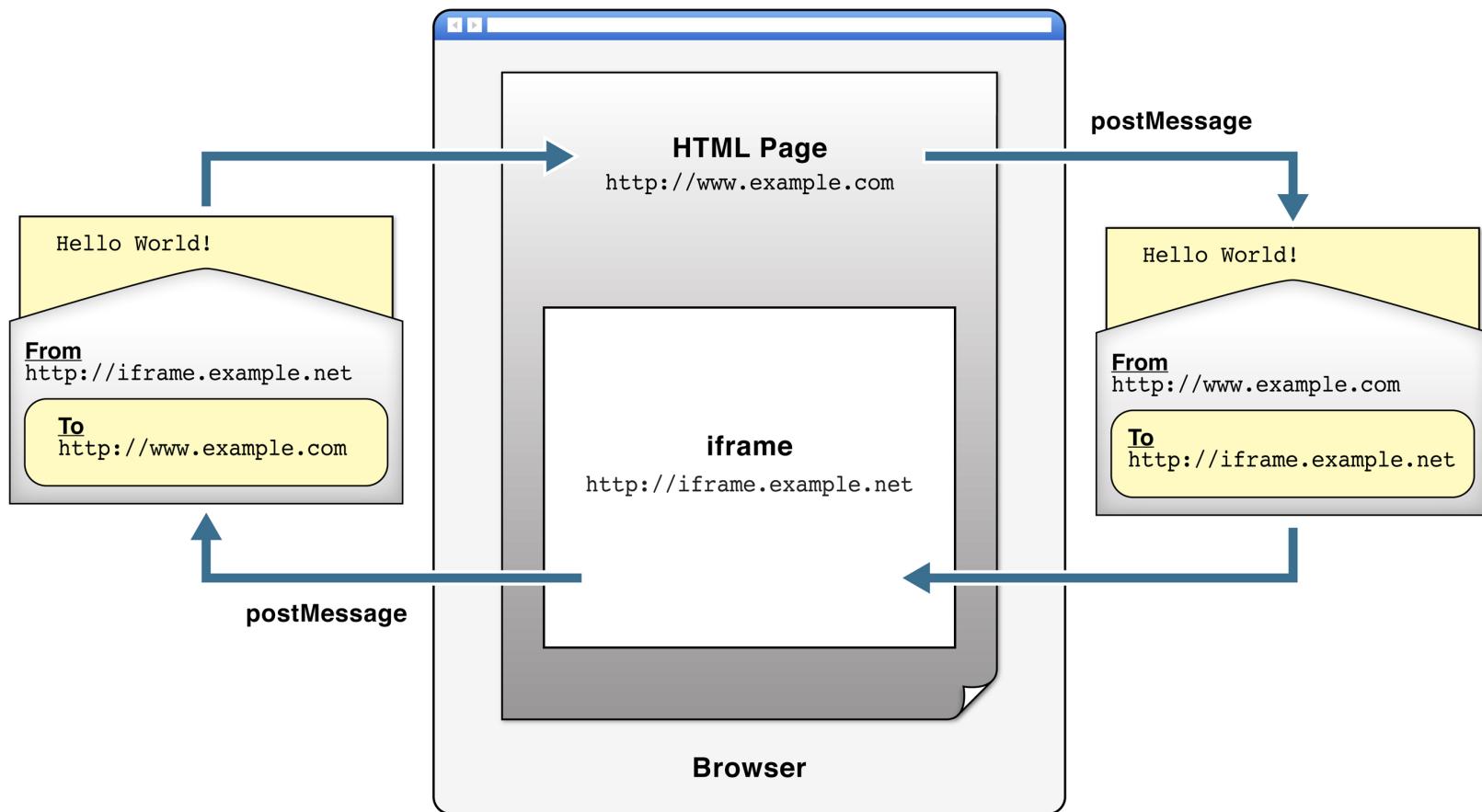
Network technologies

XHR level 2

SSE

WebSocket

Cross Document Messaging



Using PostMessage

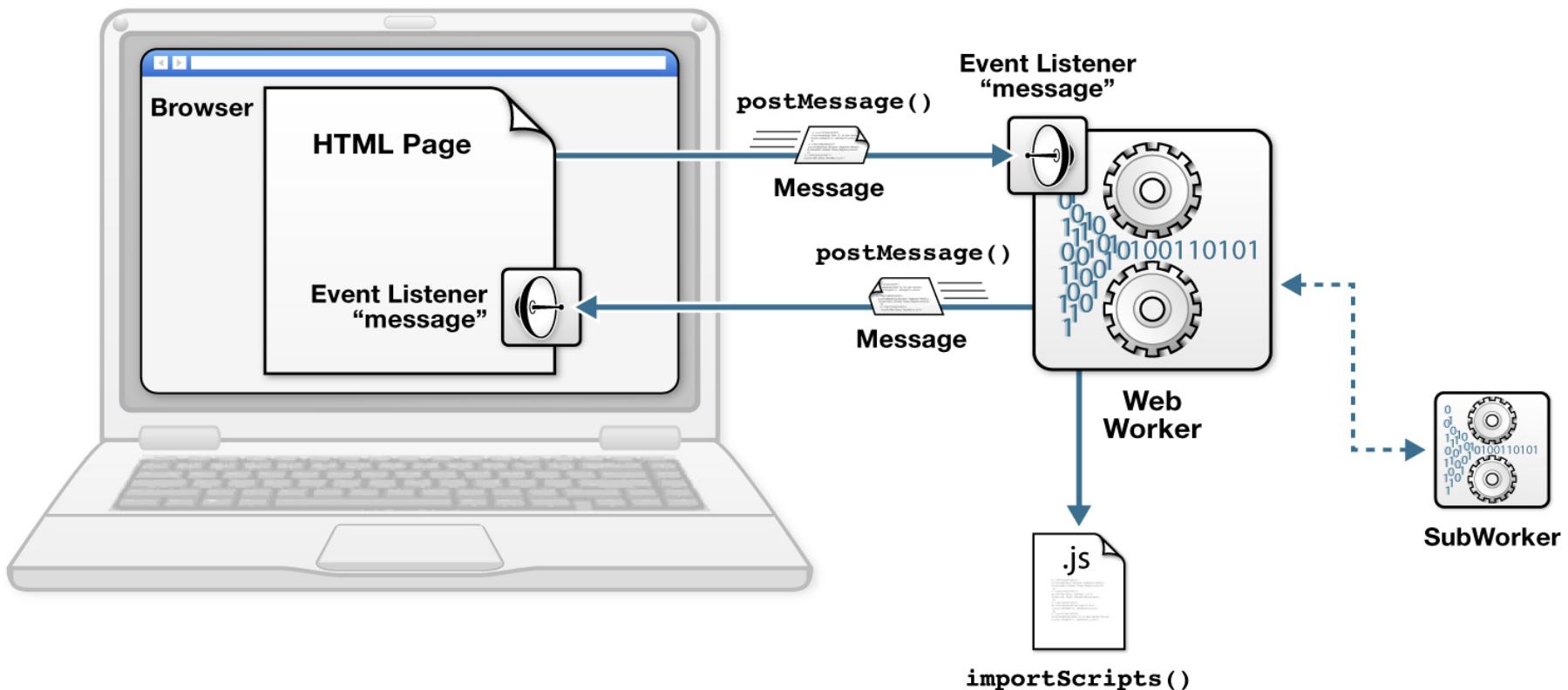
- Post to frame:

```
myFrame.contentWindow.postMessage  
('Hello', 'http://www.example.com/');
```

- Receive from frame:

```
window.onmessage = function (e) {  
    if (e.origin == "http://www.example.com") {  
        processMessage(e.data);  
    }  
    // ignore message if origin not recognized  
}
```

Web Workers



Web Workers sample

```
//Check if Web Workers are supported
if (typeof(Worker) !== "undefined") {
    var worker = new Worker("echoWorker.js");

    worker.onmessage = function(e) { ... }

    worker.onerror = errorHandler;

    worker.postMessage("start_work");

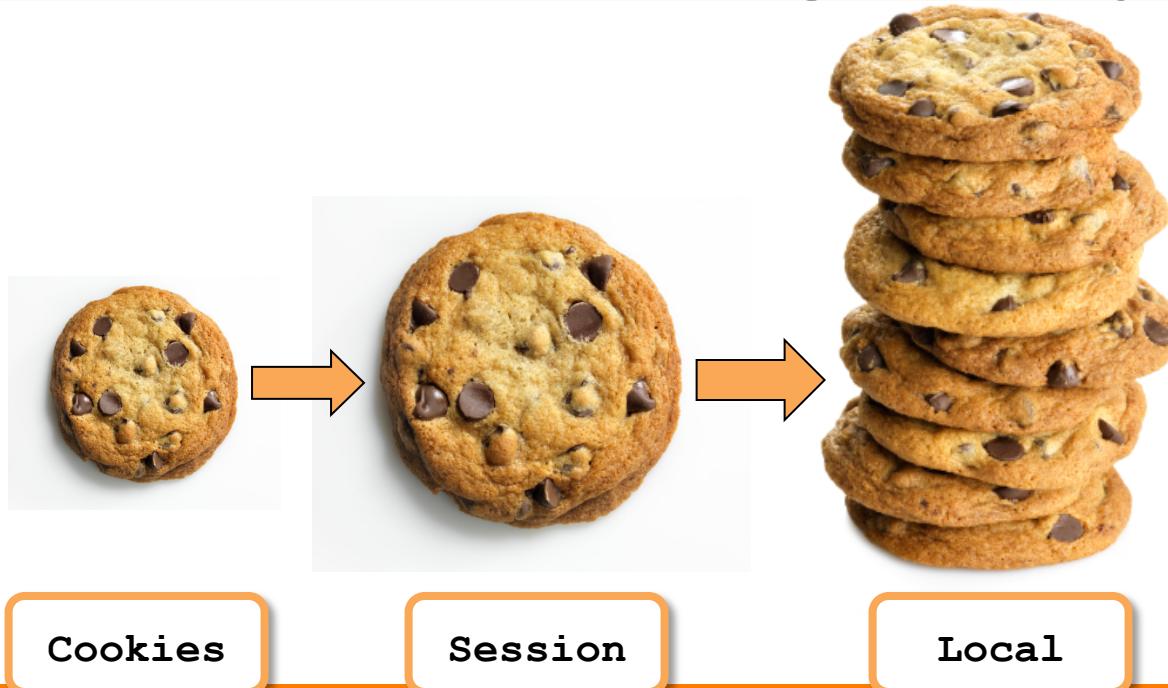
    worker.terminate();
}
```

Web Workers

- Cannot use:
 - The `window` object
 - The page or DOM API
- Can use:
 - JavaScript timing API
 - Network APIs

Web Storage

sessionStorage	localStorage
Deleted when window/tab closed.	Persist across window closure.
Visible in single window/tab.	Shared for every window or tab @ same origin.



Using Web Storage

```
window.sessionStorage.setItem('myFirstKey',  
    'myFirstValue');  
sessionStorage[key] = value;  
sessionStorage.key = value;  
  
window.sessionStorage.getItem('myFirstKey');  
sessionStorage[key];  
sessionStorage.key;  
  
for (var i=0; i < sessionStorage.length; i++) {  
    var key = sessionStorage.key(i);  
}  
  
sessionStorage.remove(key);
```

Storage Events

```
window.onstorage = function (e) {  
    var msg = "key:" + e.key + ", newValue:"  
        + e.newValue + ", oldValue:" + e.oldValue  
        + ", url:" + e.url + ", storageArea:"  
        + e.storageArea;  
    alert(msg);  
}  
}
```

Encoding for storage: JSON

“JavaScript Object Notation”

- String version of a Javascript Object:

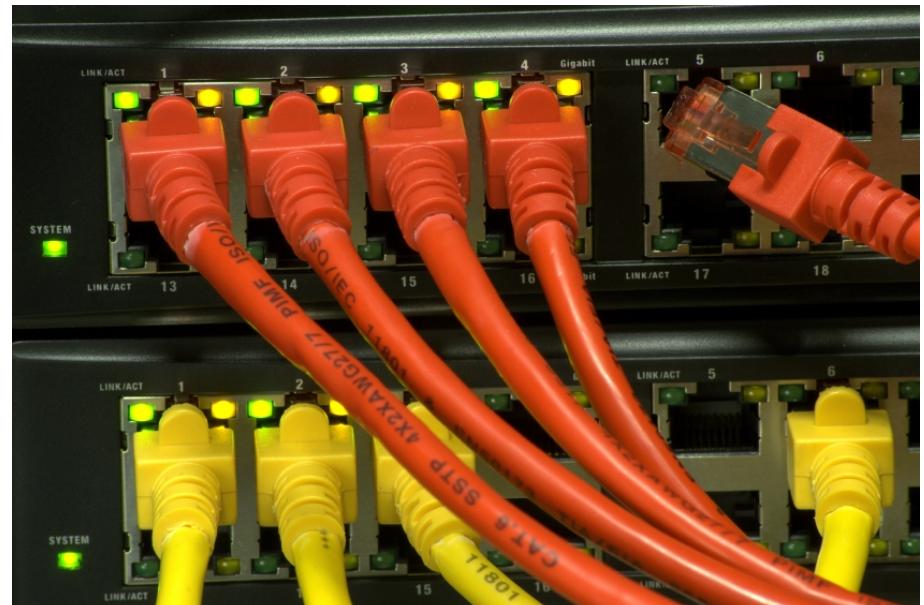
- { "stocks": [
 { "company": "AT&T", "ticker": "T",
 "price": 12.345 }] }
 - Strings, Arrays, Objects, Numbers, Booleans, Null
 - All keys must be quoted
 - All values have keys
 - Strings use double-quotes

Other storage: Databases

- [WebSQL](#) (inactive)
 - Javascript API for SQLite
 - *Browsers: WebKit (Safari, Chrome), Opera*
- [IndexedDB](#)
 - Non-relational object store (tables of objects)
 - Can define iteration order
 - *Browsers: Firefox, Chrome, IE 10*

Offline Web Applications

- Use complete web sites in offline mode
- Cache pages that have not been visited yet
- Online and offline detection



Enabling offline mode

```
<!DOCTYPE html>
<html manifest="offline.appcache">
  <head>
    <title>HTML5 Application Cache</title>
```

- Reference the manifest file:
 - Add **manifest** attribute to **html** element
 - MIME type: **text/cache-manifest**

The Manifest File

CACHE MANIFEST

```
# manifest version 1.0.1
```

```
# Files to cache
```

```
index.html
```

```
cache.html
```

```
html5.css
```

```
image1.jpg
```

```
img/foo.gif
```

```
http://www.example.com/styles.css
```

```
# Not in the app cache, fetch from network when on-line
```

NETWORK:

```
network.html
```

```
# Fallback content
```

FALLBACK:

```
/ fallback.html
```

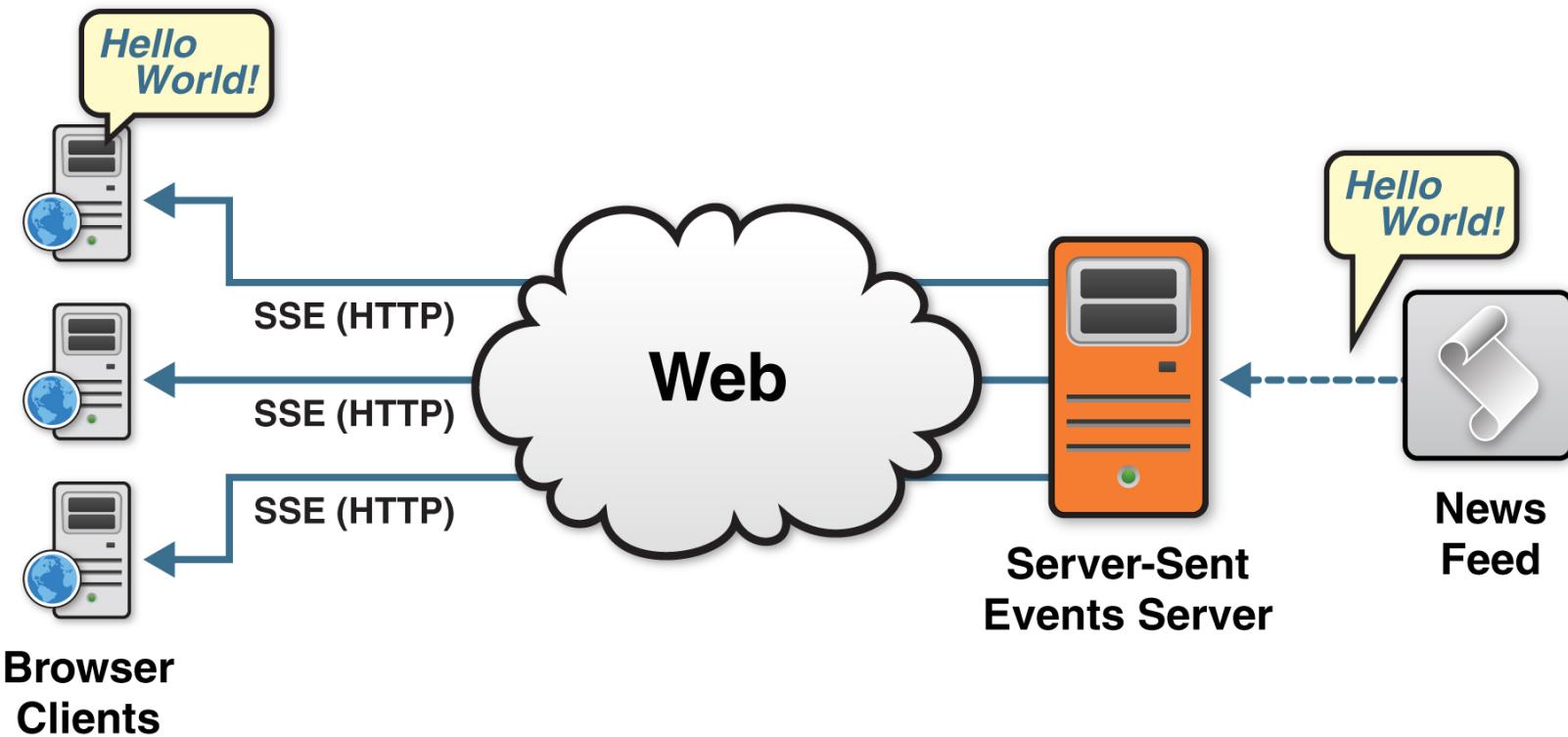
Online and Offline Events

```
window.ononline = function(e) { ... }
```

```
window.onoffline = function(e) { ... }
```

- Detect going on-line and off-line
 - Store user actions in local storage when off-line?

Server-Sent Events (e.g. COMET)

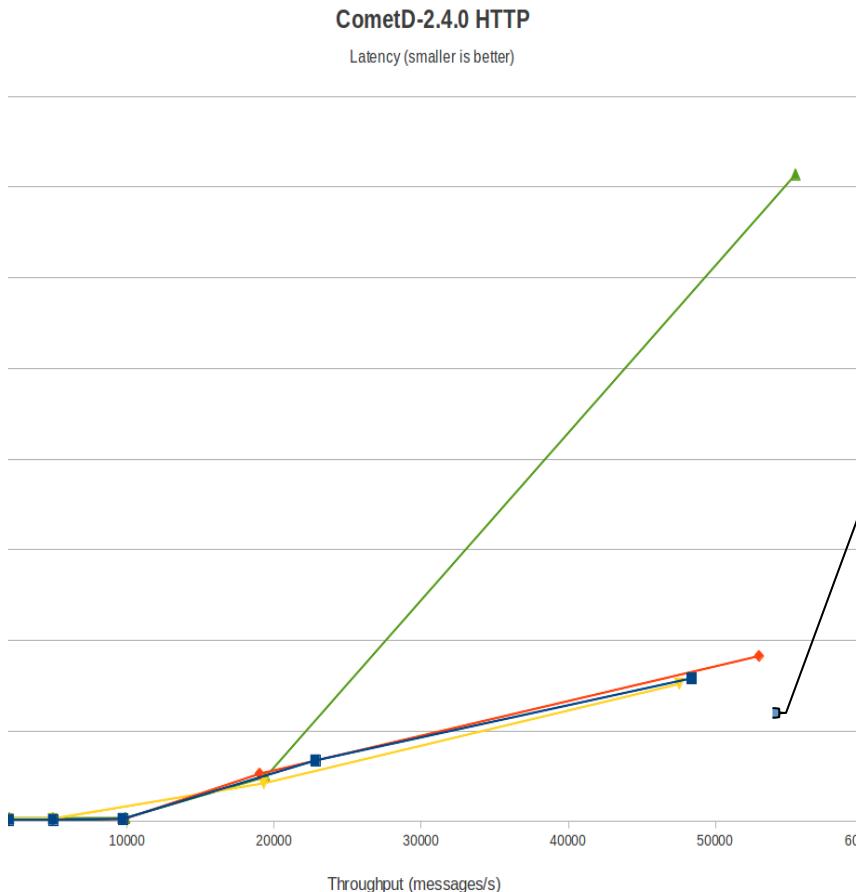


HTML5 WebSocket

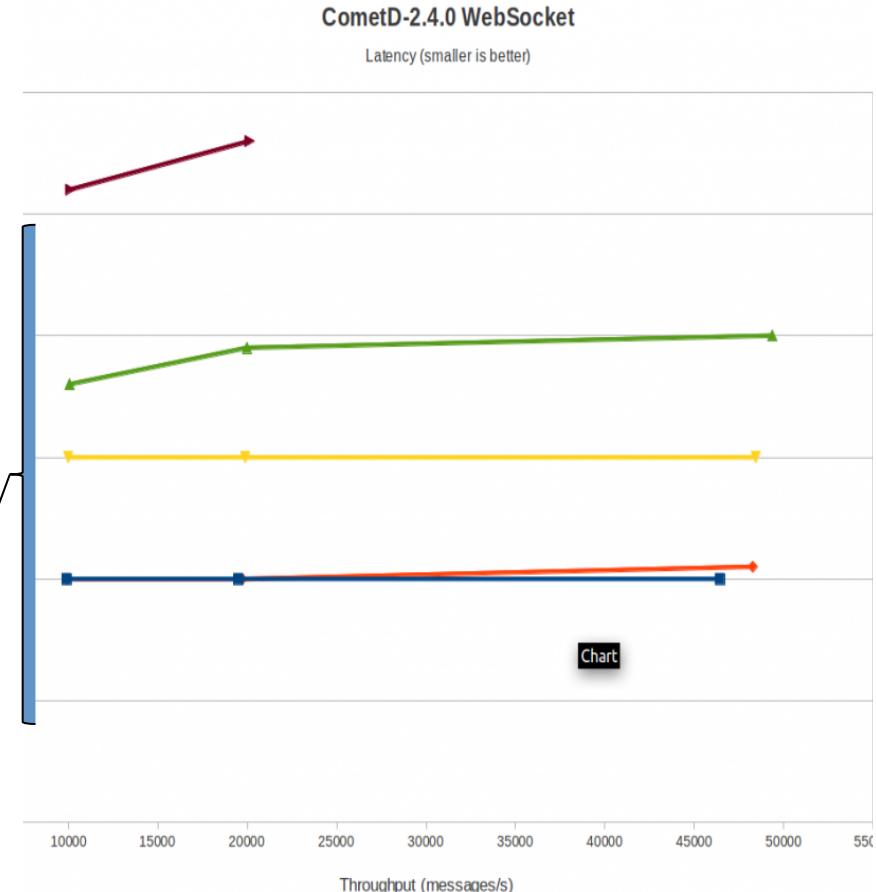
- Cross-web communications w/ remote host
 - ✓ Full-duplex (bi-directional), single socket
 - ✓ Shares port with existing HTTP content
 - ✓ Traverses firewalls and proxies
 - ✓ ws:// and wss://
- IETF Protocol and W3C API (Javascript)

Faster networking via WebSocket

Using Comet



Using WebSocket





Coping with Legacy browsers

Coping with older browsers

Testing browser readiness

- Browser test: html5test.com
- Browser support matrix: caniuse.com
- Mobile browser support matrix: mobilehtml5.org

Polyfills and Emulation:

- Supply HTML5 features to older browsers
- Complete list at: tinyurl.com/polyfills

Styling in HTML5 works in most browsers...

Header

Subtitle

Nav

- Link 1
- Link 2
- Link 3
- Link 4
- Link 5

Article Header

HTML5 ipsum dolor nunc aut nunquam sit amet, consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna. Pellentesque odio diam, rhoncus nec congue in, Class aptent taciti sociosqu ad litora torquent per conubia nostra. Per inceptos himenaeos. Quisque feugiat, justo at vehicula pellentesque, turpis lorem dictum nunc...

Article Footer

Article Header

figure and figcaption



Article Footer

Aside

HTML5: "Lorem ipsum dolor nunc aut nunquam sit amet, consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna. Pellentesque odio diam, rhoncus nec congue in, pellentesque sed tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Quisque feugiat, justo at vehicula pellentesque, turpis lorem dictum nunc."

Footer

...Except in older IE

The screenshot shows a web page with the following structure:

- Header:** Contains "Header" and "Subtitle".
- Nav:** Contains "Link 1 Link 2 Link 3 Link 4 Link 5".
- Article Header:** Contains "Article Header".
- Text:** Contains a long block of placeholder text.
- Article Footer:** Contains "Article Footer".
- Image:** A photograph of a rocky mountain landscape with trees.
- Caption:** "figure and figcaption".
- Article Footer:** Contains "Article Footer".
- Text:** Contains another block of placeholder text.
- Footer:** Contains "Footer".

A context menu is open in the top right corner of the browser window, listing "Allow Blocked Content...", "What's the Risk?", and "Information Bar Help". An orange callout bubble labeled "I.E. <9" points to the menu.

Fixing IE (6, 7, 8)

code.google.com/p/html5shiv/

```
<head>
  <!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/
  trunk/html5.js"></script>
  <! [endif]-->
</head>
```

Now it works

Header

Subtitle

Nav

- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)

Article Header

Lorem ipsum dolor **HTML5 nunc aut nunquam** sit amet, consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna. Pellentesque odio diam, rhoncus nec congue in, Class aptent taciti sociosqu ad litora torquent per conubia nostra. Per inceptos himenaeos. Quisque feugiat, justo at vehicula pellentesque, turpis lorem dictum nunc...

Article Footer

Article Header

figure and figcaption



Article Footer

Aside

HTML5: "Lorem ipsum dolor nunc aut nunquam sit amet, consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna. Pellentesque odio diam, rhoncus nec congue in, pellentesque sed tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Quisque feugiat, justo at vehicula pellentesque, turpis lorem dictum nunc."

Footer

Maintaining Compatibility

- **html5shiv**
 - Creates the new HTML5 DOM elements in Internet Explorer so that styles can be applied to them

<http://code.google.com/p/html5shiv/>
- **Modernizr**
 - JavaScript library that checks support for HTML5 and CSS3 features (includes the html5shiv)
 - Feature detection, not browser detection
 - Includes html5shiv

<http://www.modernizr.com/>
- **Polyfills**
- **~~Google Chrome Frame~~**

Modernizr

The screenshot shows the official website for Modernizr. At the top, there's a dark header bar with the Modernizr logo and navigation links for DOWNLOAD, DOCUMENTATION, RESOURCES, and NEWS. Below the header, a quote from Andy Clarke is displayed: "Modernizr is an essential part of my toolkit of files." — Andy Clarke, *Stuff & Nonsense*. The main content area has a pink background with white diagonal stripes. On the left, there's a summary of what Modernizr is, followed by a section titled "Why use Modernizr?". This section contains two columns of text: one about the challenges of using new web technologies and another about how Modernizr helps. To the right of this is a large callout box for "Download Modernizr 2". It includes a link to "View documentation", a "DEVELOPMENT" button (with a wrench icon) leading to an uncompressed development version (42 Kb), and a "PRODUCTION" button (with a checkmark icon) leading to a tool for configuring your build. Below this is a section titled "Get started with Modernizr" which provides tips for using the library effectively. A sidebar on the right lists several resources and links for further reading.

Modernizr is an open-source JavaScript library that helps you build the next generation of HTML5 and CSS3-powered websites.

Why use Modernizr?

Taking advantage of the new capabilities of HTML5 and CSS3 can mean sacrificing control over the experience in older browsers. Modernizr 2 is your starting point for making the best websites and applications that work exactly right no matter what browser or device your visitors use.

Thanks to the new Media Query tests and built-in [YepNope.js](#) micro-library as `Modernizr.load()`, you can now combine feature detection with media queries and conditional resource loading. That gives you the power and flexibility to optimize for every circumstance.

Check out the [full list of features](#) that Modernizr detects, or learn more about [conditional resource loading with Modernizr](#).

Download Modernizr 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](#)
- [wiki] [The Undetectables: features that cannot be detected](#)
- [wiki] [Cross-browser Polyfills](#)

Also check out our [Resources section](#).

Tip: use [haz.io](#) to quickly test your current browser's features.

Modernizr

<http://www.modernizr.com/>

HTML

```
<html class="no-js">
  <head>
    <link rel="stylesheet" ... >
    <script src="/js/modernizr-1.5.min.js"></script>
```

CSS

```
.no-js .glossy, .no-cssgradients .glossy {
  background: url("images/glossybutton.png");
}

.cssgradients .glossy {
  background-image: linear-gradient(top, #555, #333);
}
```

Polyfills (emulation)

- Available for most HTML5 features
- Sometimes use plugins (e.g. Flash)
- Complete list: <http://tinyurl.com/polyfills>

Javascript

```
Modernizr.load({
  test: Modernizr.geolocation,
  yep : 'geo.js',
  nope: 'geo-polyfill.js'
});
```



Wrapping up

What HTML5 Changes

- ✓ Greater separation of content and styling
- ✓ Greater emphasis on client-side interactivity
- ✓ Wider range of tools available (many open-source)
- ✓ Greater need for modern software engineering techniques

Key Skills for Modern Devs

- ✓ Separating content from style
- ✓ Writing clean “semantic” markup
- ✓ CSS layout techniques
- ✓ Adapting to browser differences
- ✓ Developing & debugging interactive code

Learn More...

- Apress book: *Pro HTML5 Programming*
 - (Peter Lubbers, Brian Albers, & Frank Salim)
<http://www.prohtml5.com/>
- Kaazing Corporation—Training:
 - [E-mail us: training@kaazing.com](mailto:training@kaazing.com)
 - <http://tech.kaazing.com/training/>

Resources::Books

- Modern web design
 - [Responsive Web Design](#), Ethan Marcotte
 - [Adaptive Web Design](#), Aaron Gustafson
 - [Hardboiled Web Design](#), Andy Clarke
 - [Mobile First](#), Luke Wrobleowski
 - [Pro HTML5 Programming](#), Peter Lubbers, Brian Albers, Frank Salim
 - [Handcrafted CSS: More Bulletproof Web Design](#), Dan Cederholm, Ethan Marcotte

Resources::Books

- Javascript
 - [*Javascript: The Good Parts*](#) by Douglas Crockford
 - Eloquent Javascript @ eloquentjavascript.net
- Foundations
 - [Head First Web Design](#), Ethan Watrall, Jeff Siarto
 - [Web Design in a Nutshell](#), Jennifer Niederst Robbins (N.B. this is a reference book)
 - [CSS: the definitive guide](#), Eric Meyer

Resources::Web

- Javascript
 - Daily JS: dailyjs.com
 - JavaScript, JavaScript: javascriptweblog.wordpress.com
 - Thomas Fuchs: mir.aculo.us
 - YUI Blog: www.yuiblog.com/blog
 - [Javascript Garden](#)
 - [JSLint.com](#)

Resources::Web

- Performance
 - High Performance Web Sites: www.stevesouders.com/blog

THANKS!

E-mail: richard.clark@kaazing.com

Twitter: [@rdclark](https://twitter.com/@rdclark)

LinkedIn: Richard Clark





KAAZING