



A JavaScript Outlook:

How the Client gets more powerful

Overview

A JavaScript ~~Outlook~~:

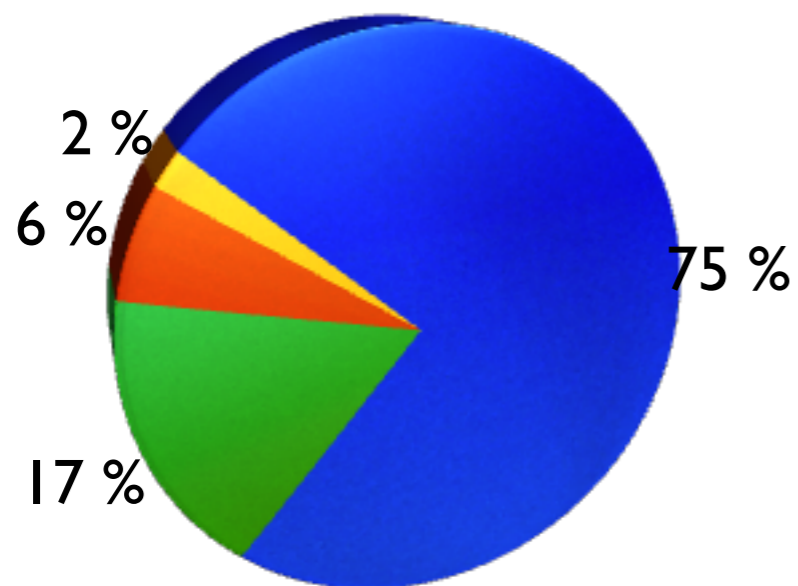
How the Client gets more powerful

Agenda

- State of affairs
- Here and now
 - Canvas
 - CSS
 - Storage
 - Worker-Threads
 - Frameworks
- The future ...

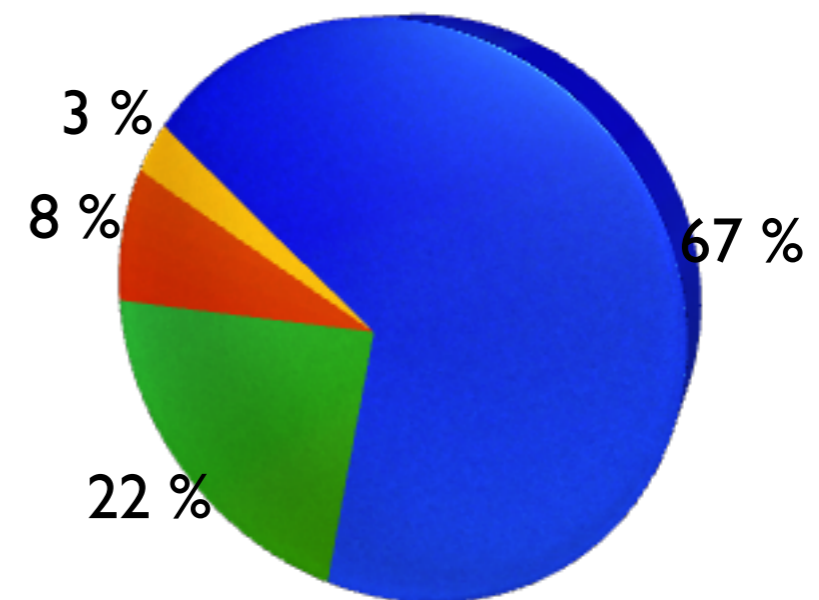
State of affairs

Browsers overall - 02/2008

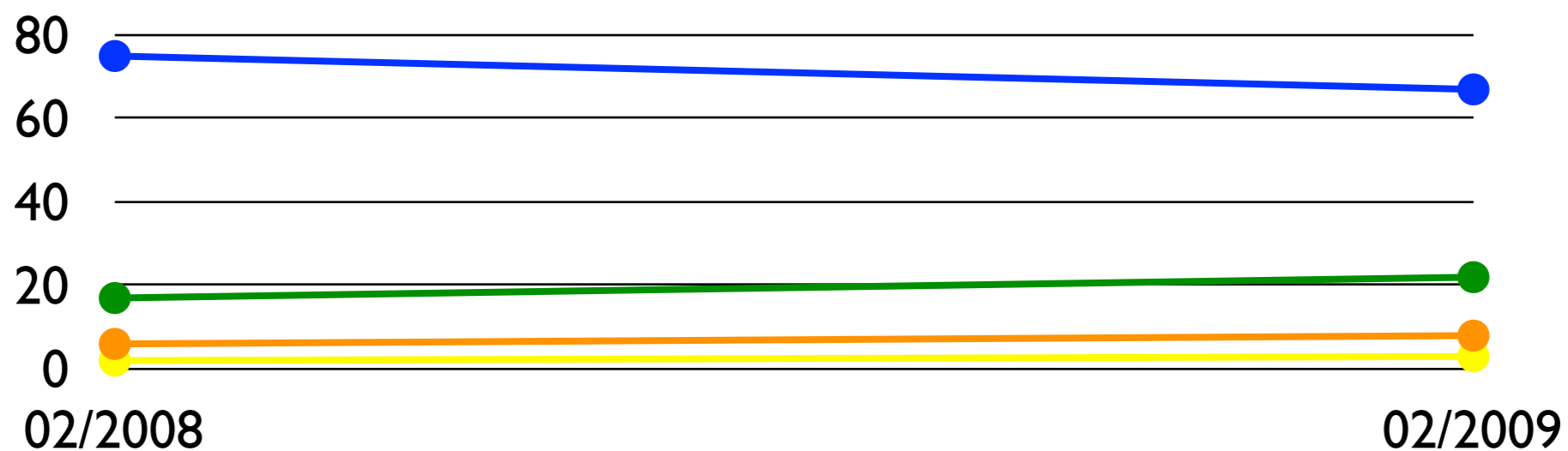


● Internet Explorer ● Firefox ● Safari ● Other

Browsers overall - 02/2009

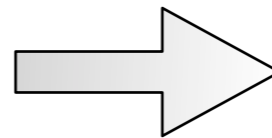
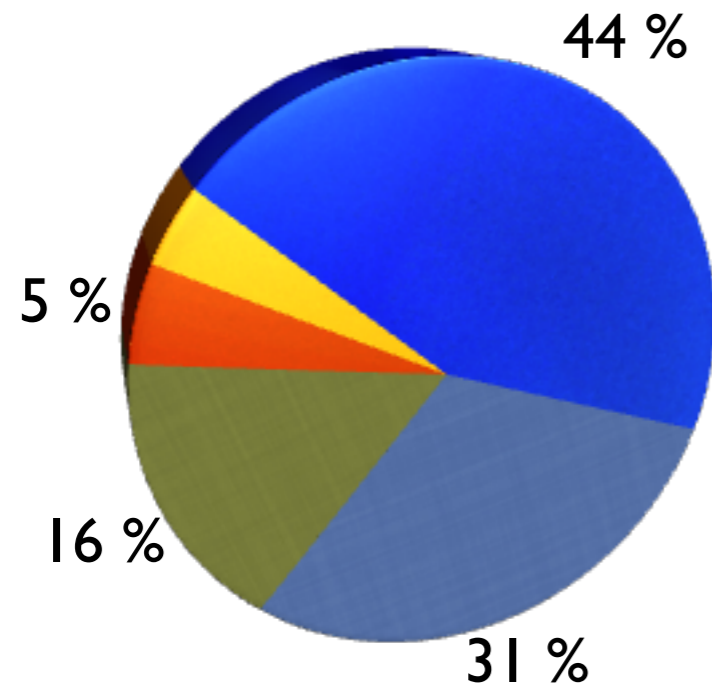


● Internet Explorer ● Firefox ● Safari ● Other

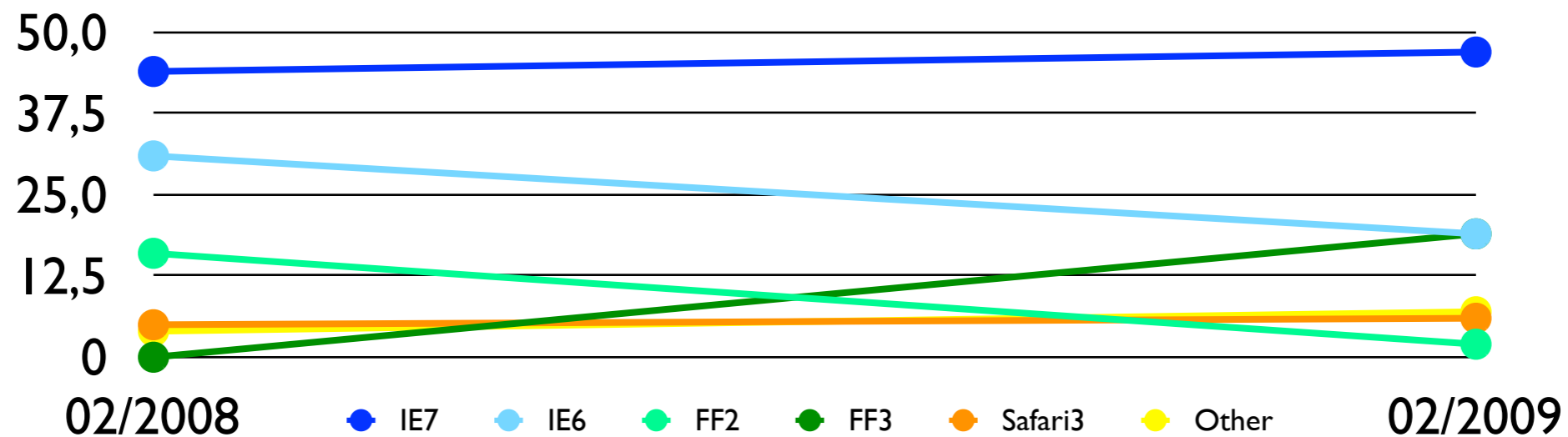
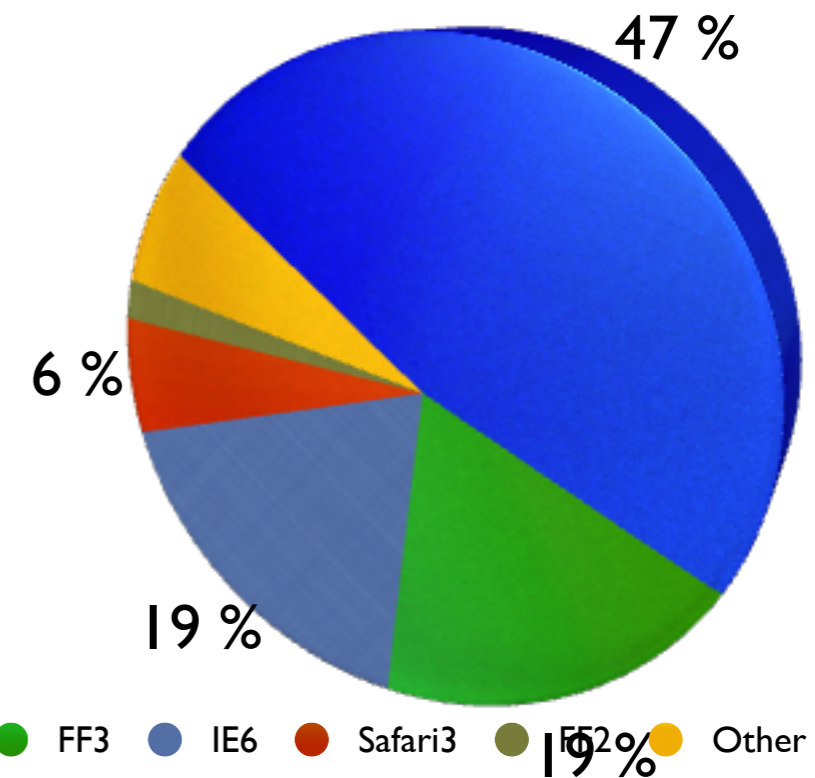


State of affairs

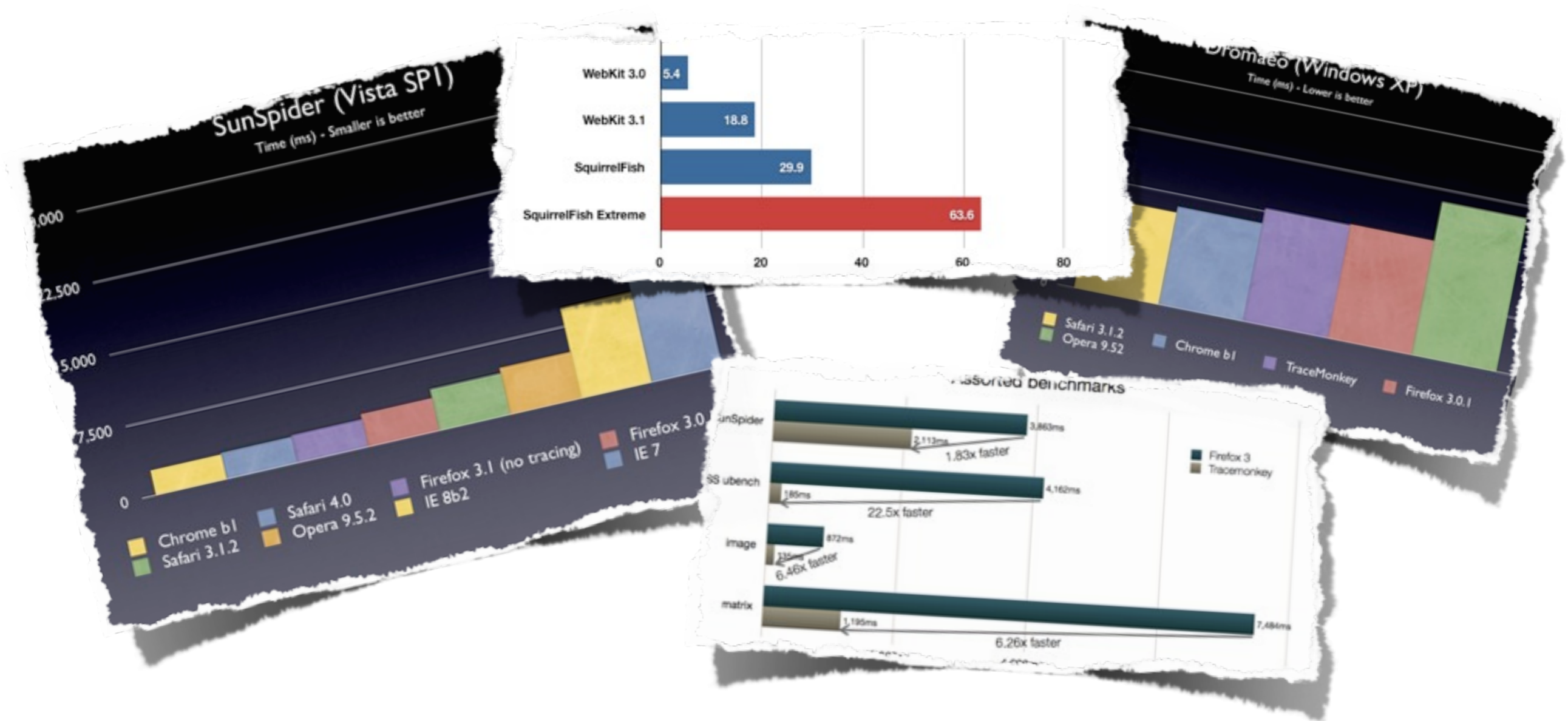
Browser versions - 02/2008



Browser versions - 02/2009



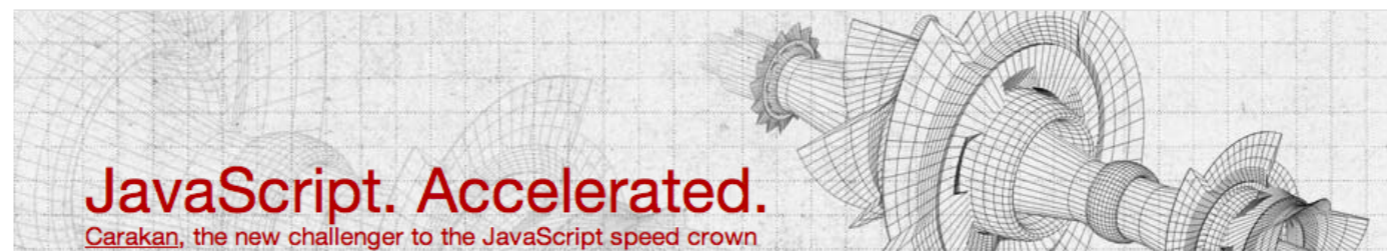
State of affairs



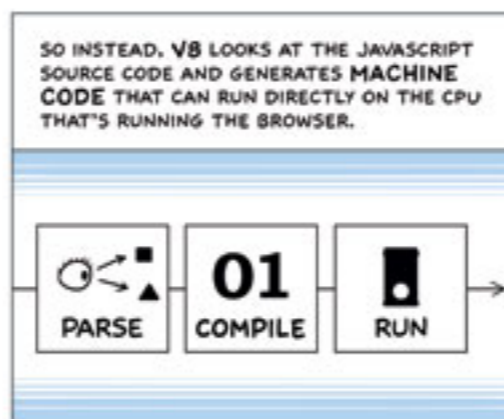
- JavaScript gets faster and faster

What developers can do with current browsers

The here and now!



Firefox 3.5 for developers



te that this
ased. See Firefox

web browser, Safari outraces Firefox,
rome. On even the most demanding
fari delivers blazingly fast performance
most advanced rendering technologies.

ine, for example, Safari executes
i faster than Internet Explorer 7 and
than Firefox 3 based on performance in
ark tests: iBench and SunSpider.

WHEN YOU INTERPRET ONCE AND COMPILE MACHINE CODE, THEN
THAT CODE IS YOUR REPRESENTATION OF THE JAVASCRIPT SOURCE
CODE AND IT DOESN'T NEED TO BE INTERPRETED, IT JUST RUNS.



In addition to superior JavaScript performance, Safari offers top-flight HTML performance — the best on any platform — loading pages 3 times faster than Internet Explorer 7 and almost 3 times faster than Firefox 3.

A small performance chart with two bars. The first bar is blue and labeled "1.06 s". The second bar is yellow and labeled "1.38 s".

Blazing performance
How does Safari performance compare to other browsers on the Mac and Windows?
[View the performance charts](#)

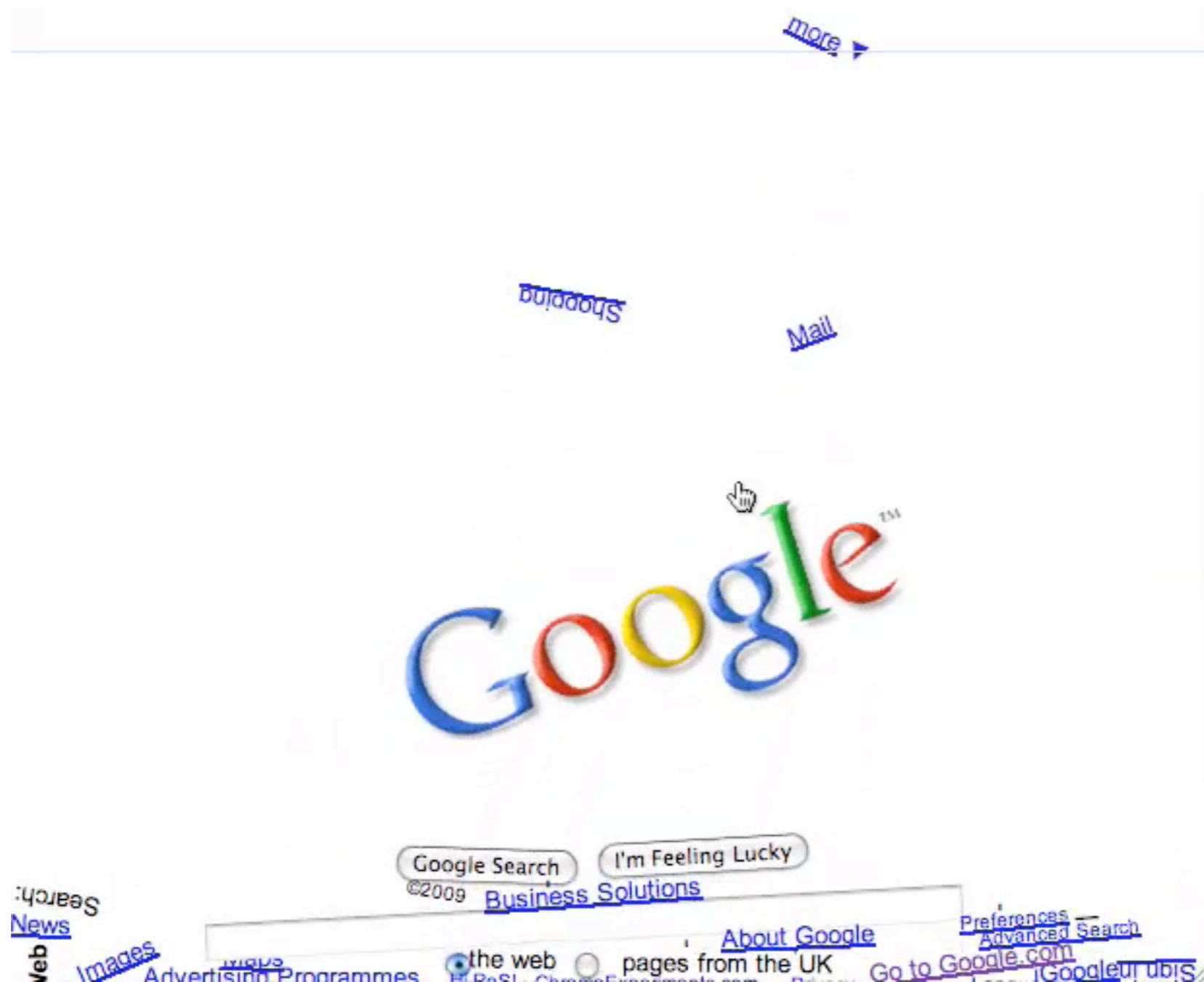
CSS transformations & animations

A screenshot of the Safari 4 welcome screen. The text "Welcome to Safari 4" is displayed in a large, white, sans-serif font, centered on a black background. Below the text, there is a faint, semi-transparent reflection of the text, creating a 3D effect.

Welcome to Safari 4

<http://www.apple.com/safari/welcome/>

CSS transformations & animations

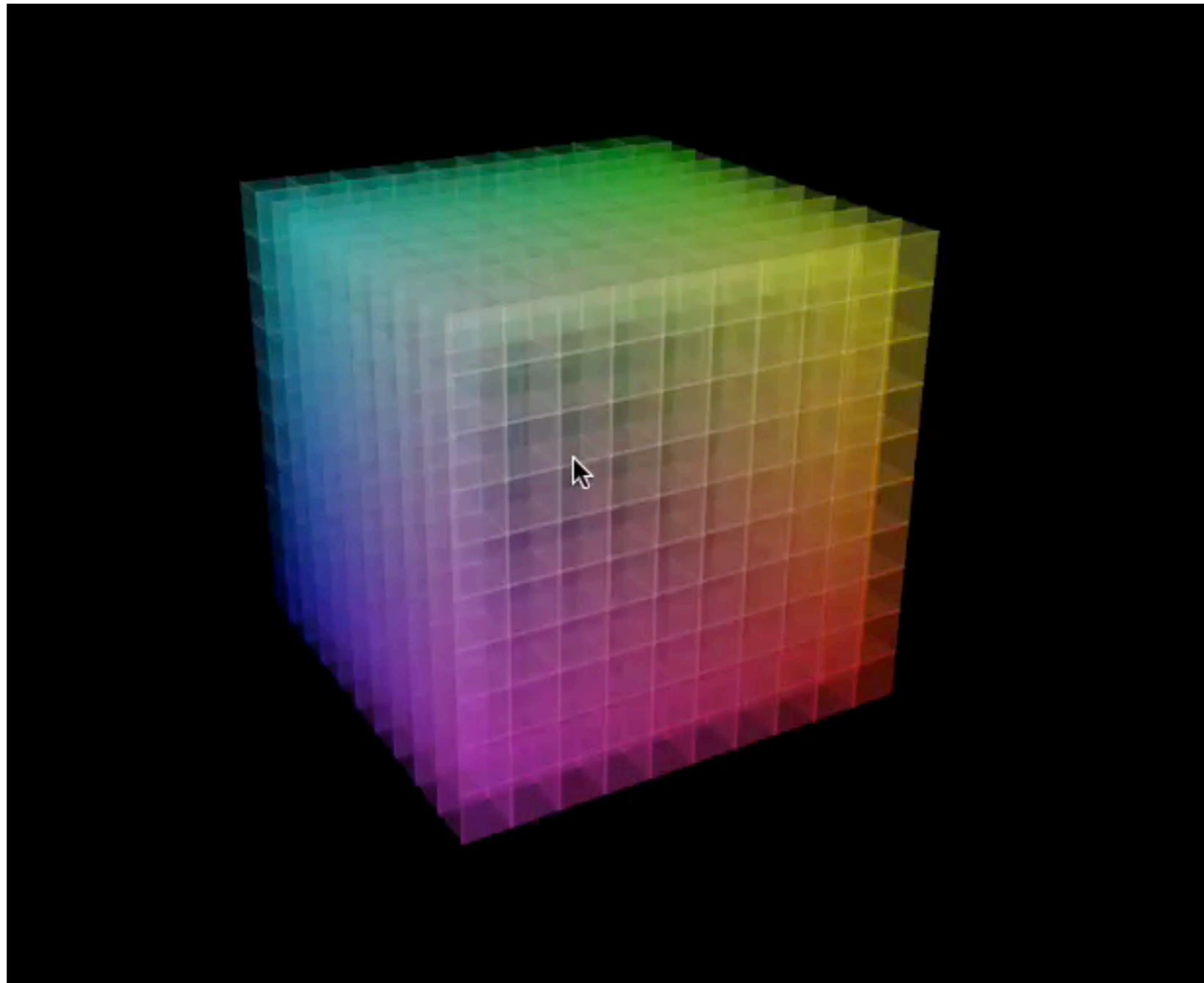


<http://www.chromeexperiments.com/detail/gravity/>

CSS transformations & animations

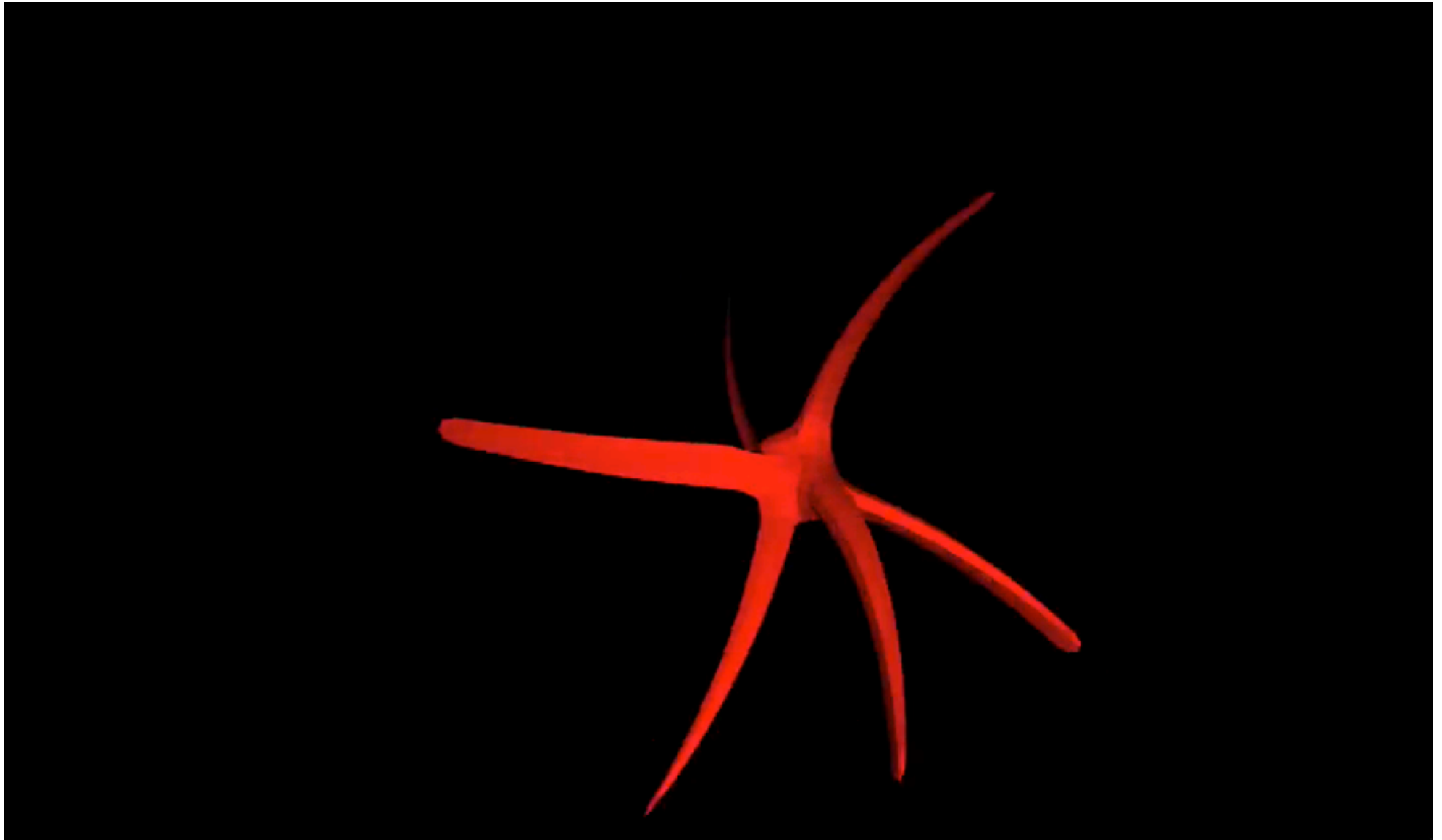
- New technologies to manipulate and animate HTML object with CSS
- Could replace JS calculated animations
- Supports 2d & 3d transformations
 - Scale
 - Translate
 - Rotate
 - Skew
- Supports animations...
 - ... for classical CSS properties like position/opacity
 - ... for the new transformations

Canvas



<http://www.chromeexperiments.com/detail/colorscube/>

Canvas

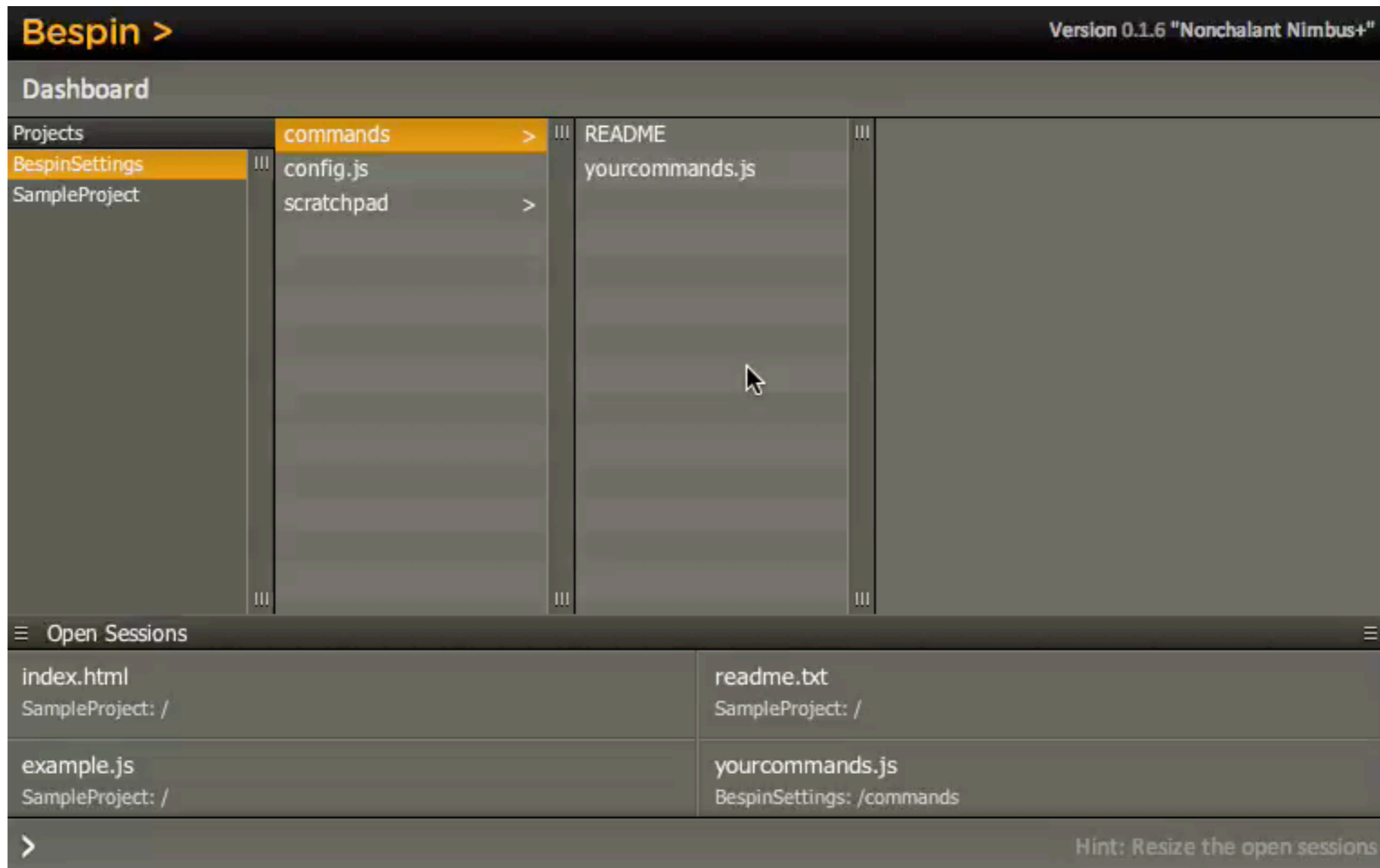


<http://www.chromeexperiments.com/detail/monster/>

Canvas

- Canvas is a HTML5 2d drawing element
- JavaScript drawing API
- Allows to render anything pixel by pixel
- Already available in every browser (besides IE)

Canvas



<https://bespin.mozilla.com/>

Canvas

- Bespin
 - Mozilla Labs online text editor
 - Uses Canvas for nearly everything
 - Text rendering
 - Blinking cursor
 - UI rendering
 - Scrollbars
 - Panels
 - List views
 - ...

HTML video element

which gives native playback controls, or the specification are there yet. The current implementation supports all formats that QuickTime supports, including installed 3rd party codecs.

The example below uses the 'poster' attribute of the <video> element to display an initial image before the video is loaded, progress events to track loading, and play/pause/ended events to make the overlay button reflect the video's state.



You can follow any responses to this entry through the [RSS 2.0](#) feed. Both comments and pings are currently closed.

13 Responses to "HTML5 Media Support"

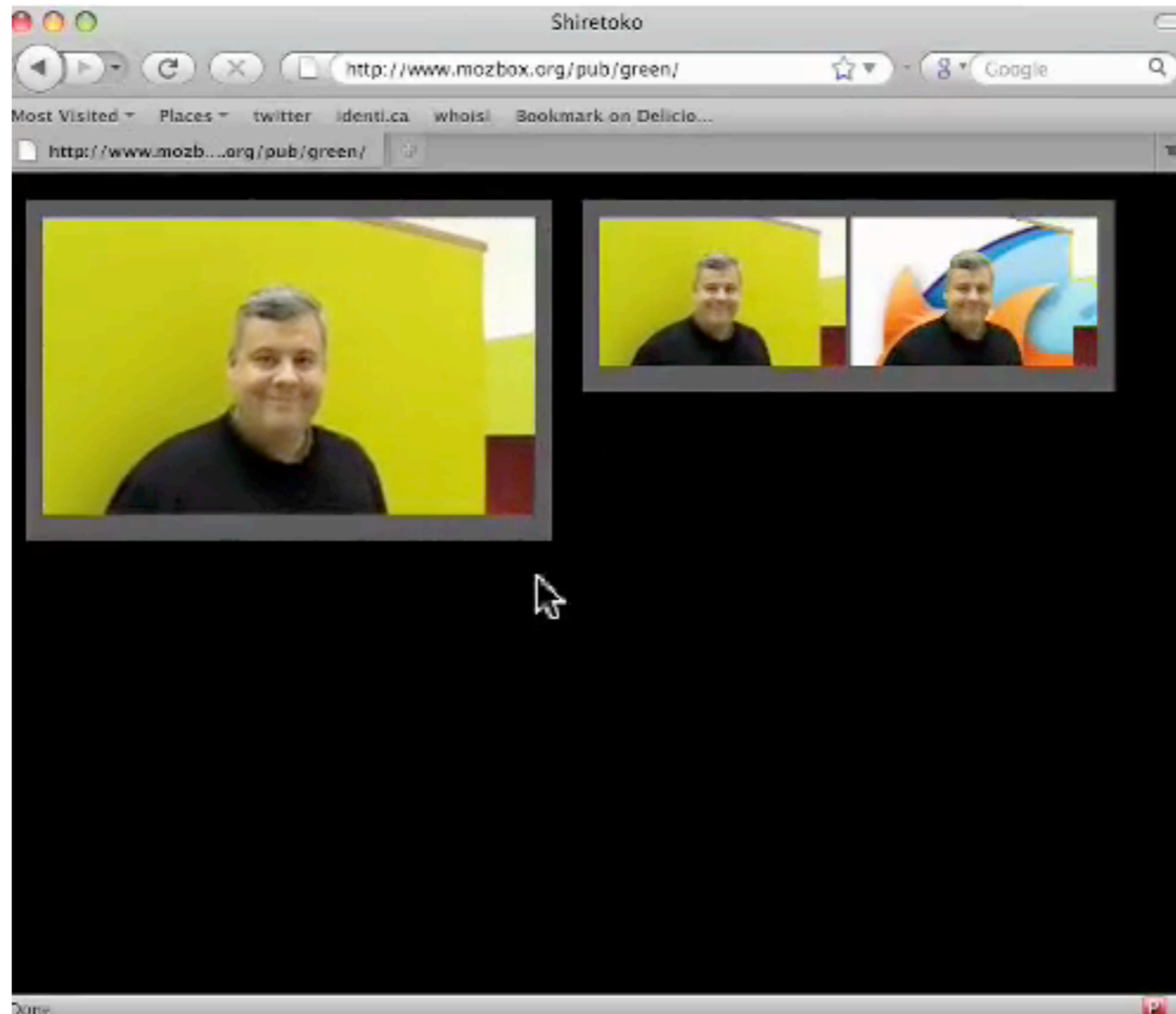
Hendrik.Runte Says:

November 12th, 2007 at 2:48 pm

Thank you so much! This object-and-embed-stuff has been really a PITA, so far. The only workaround, a flash object (flv player), is reliable but not scriptable as your solution. Cheers!

<http://webkit.org/blog/140/html5-media-support/>

HTML video element



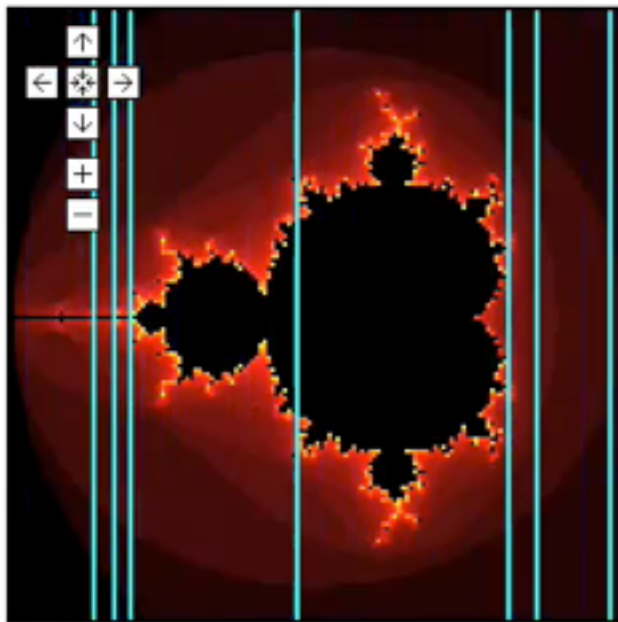
https://developer.mozilla.org/En/Manipulating_video_using_canvas

HTML video element

- Native HTML elements for audio and video
- JS API
- Canvas API
 - Allows to access video data and work with it on a pixel level

Workers

Mandelbrot Demo using Google Gears



Range in Mandelbrot set: $(-2, -1.5i) - (1, 1.5i)$

Stop

Interact with UI:

Starting.

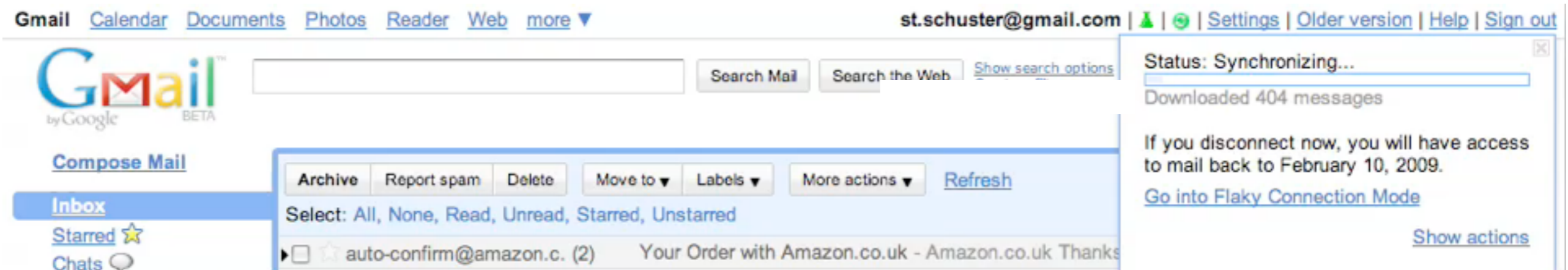
Worker 1 did column 0
Worker 3 did column 54
Worker 4 did column 127
Worker 2 did column 56
Worker 5 did column 5
Worker 1 did column 104
Worker 3 did column 23
Worker 4 did column 92
Worker 2 did column 116
Worker 5 did column 76
Worker 1 did column 66
Worker 3 did column 61
Worker 4 did column 12
Worker 2 did column 13
Worker 5 did column 49

<http://n.a.edgar.googlepages.com/mandelbrot-gg.html>

Workers

- Like multithreading for JavaScript
 - Background script execution
 - Parallel execution
 - Doesn't freeze the UI
 - Now available via Google Gears
 - Workers will be part of HTML5

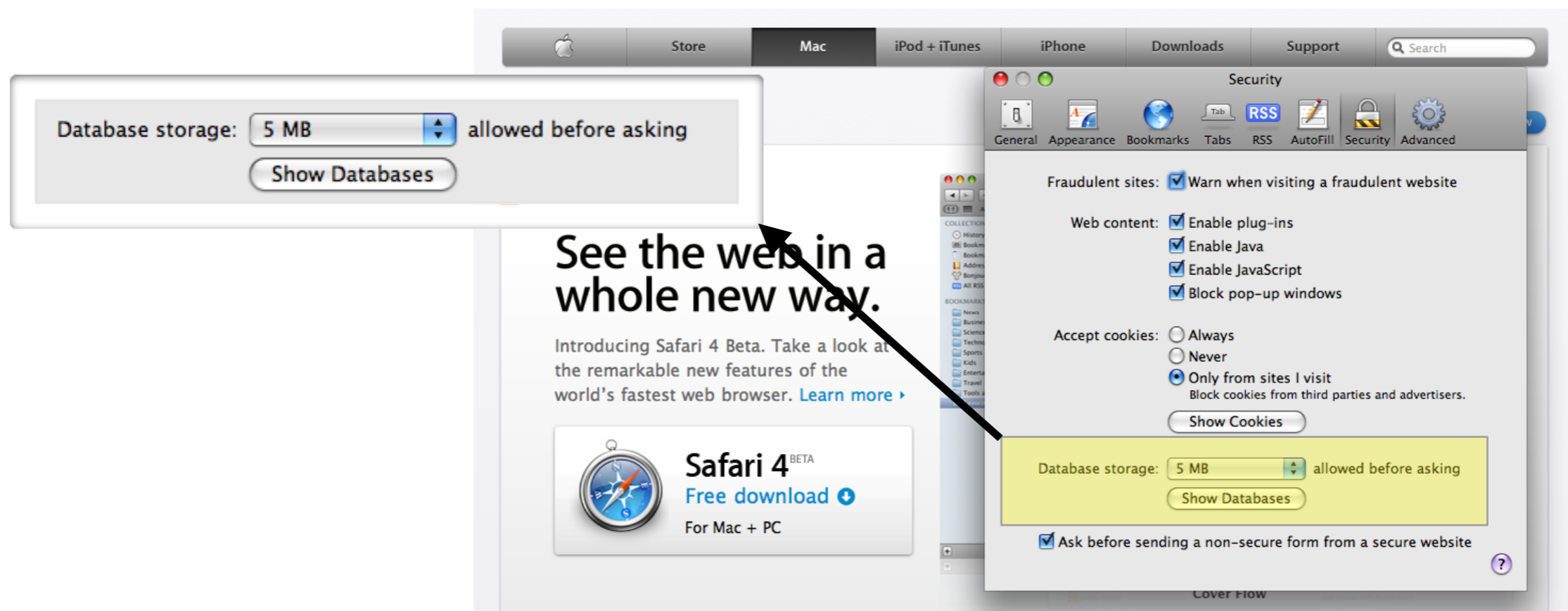
Storage



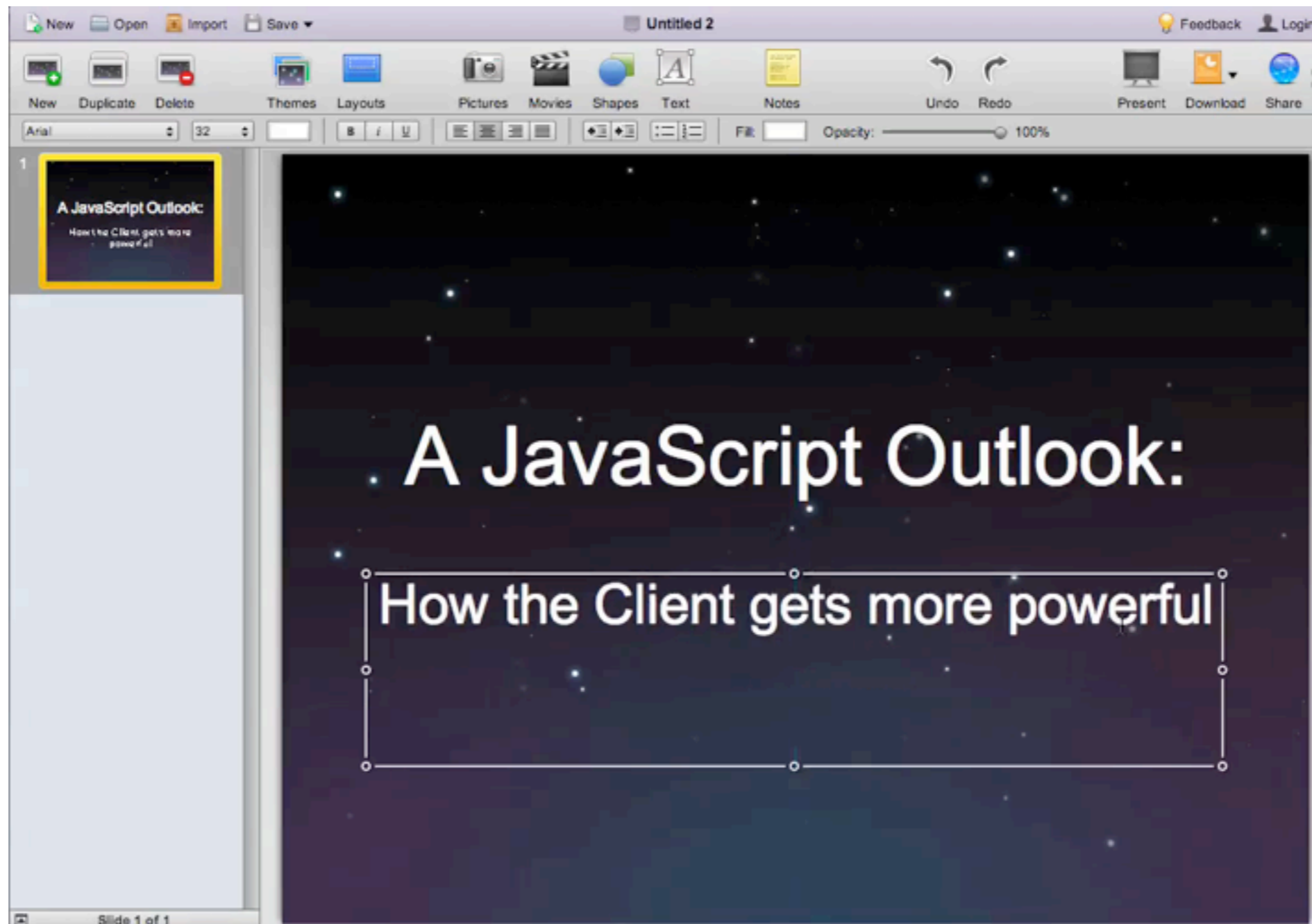
<http://mail.google.com>

Storage

- Client-Side storage for
 - JS API for a relational database
 - Introduced with Gears
 - Also available in HTML5



Frameworks



<http://280slides.com/>

Frameworks

- Objective-J
 - Object-oriented syntax enhancement for JavaScript
 - Like Objective-C for C (also same syntax)
 - JS interpreter
- Cappuccino
 - Application & UI framework written in Objective-J
 - Like Cocoa for Objective-C



<http://cappuccino.org/>

Code snippets:

```
var myPerson = [[Person alloc] init];  
[myPerson setName: "John"];  
  
[myPerson setJobTitle: "Founder" company: "280 North"];
```

Frameworks

Cappuccino allows higher level programming

“Cappuccino is built on top of standard web technologies like JavaScript, and it implements most of the familiar APIs from GNUstep and Apple's Cocoa frameworks. When you program in Cappuccino, you don't need to concern yourself with the complexities of traditional web technologies like HTML, CSS, or even the DOM. The unpleasantries of building complex cross browser applications are abstracted away for you.”

[Main Page](#) [Related Pages](#) [Classes](#) [Files](#)

[Class List](#) [Class Members](#)

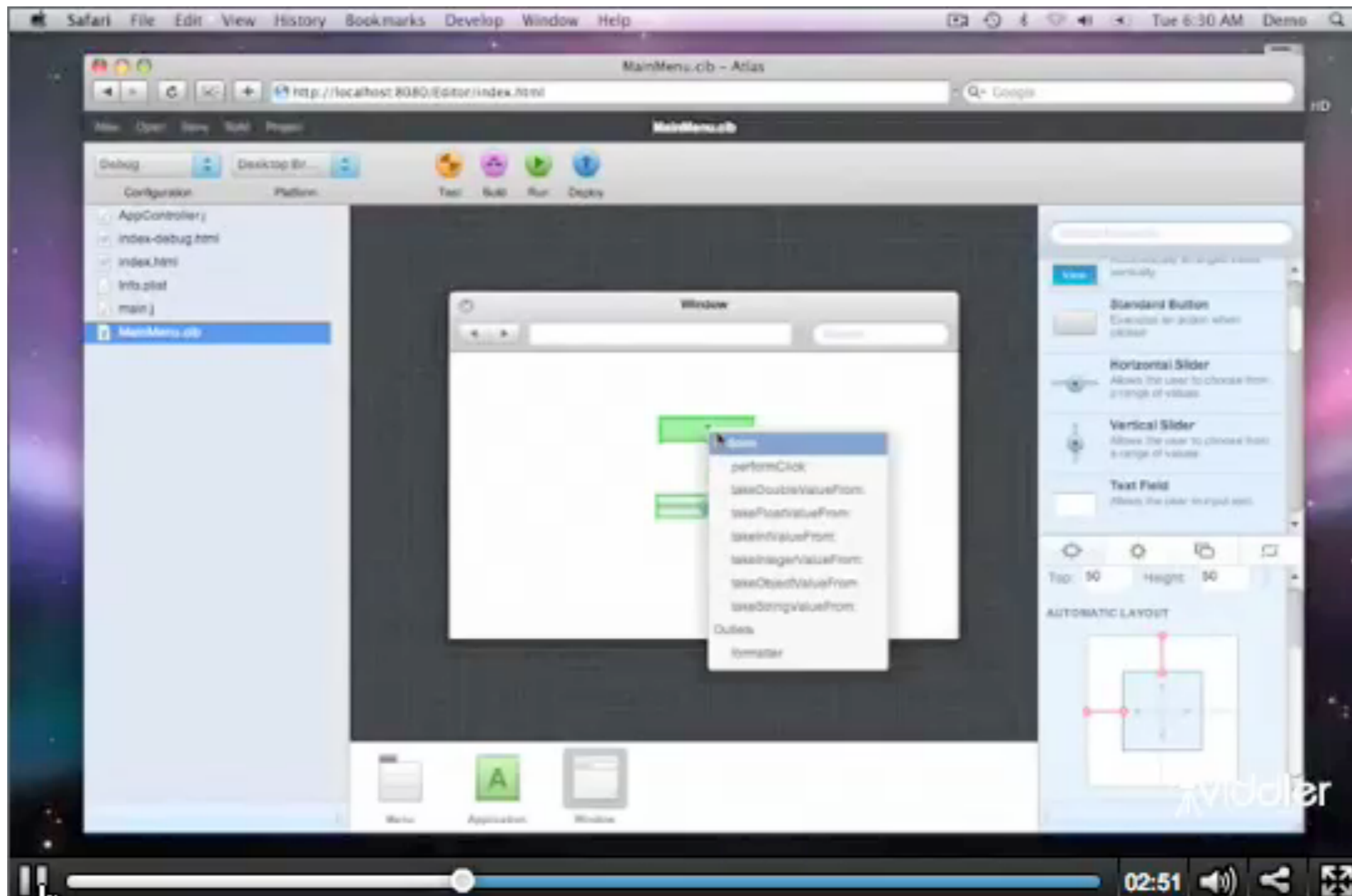
CPWindow Class Reference

List of all members.

Public Member Functions

(id)	- initWithContentRect:styleMask:	[implementation]
(id)	- initWithContentRect:styleMask:bridge:	[implementation]
(unsigned)	- styleMask	[implementation]
(CGRect)	- contentRectForFrameRect:	[implementation]
(CGRect)	- frameRectForContentRect:	[implementation]
(CGRect)	- frame	[implementation]
(void)	- setFrame:display:animate:	[implementation]
(void)	- setFrame:	[implementation]
(void)	- setFrameOrigin:	[implementation]
(void)	- setFrameSize:	[implementation]
(void)	- trackMoveWithEvent:	[implementation]
(void)	- trackResizeWithEvent:	[implementation]
(void)	- orderFront:	[implementation]
(void)	- orderBack:	[implementation]
(void)	- orderOut:	[implementation]
(void)	- orderWindow:relativeTo:	[implementation]
(void)	- setLevel:	[implementation]
(int)	- level	[implementation]
(BOOL)	- isVisible	[implementation]
(BOOL)	- showsResizeIndicator	[implementation]
(void)	- setShowsResizeIndicator:	[implementation]
(CGSize)	- resizeIndicatorOffset	[implementation]
(void)	- setResizeIndicatorOffset:	[implementation]
(void)	- setContentView:	[implementation]
(CPView)	- contentView	[implementation]
(void)	- setBackgroundColor:	[implementation]
(CPCOLOR)	- backgroundColor	[implementation]
(void)	- setMinSize:	[implementation]
(CGSize)	- minSize	[implementation]
(void)	- setMaxSize:	[implementation]
(CGSize)	- maxSize	[implementation]
(BOOL)	- hasShadow	[implementation]
(void)	- setHasShadow:	[implementation]

Frameworks



<http://280north.com/blog/2009/02/announcing-atlas/>

Frameworks

- 280 North even announced a visual Cappuccino IDE
 - Written in Cappuccino
 - Like Interface Builder for Cocoa



<http://280atlas.com/>

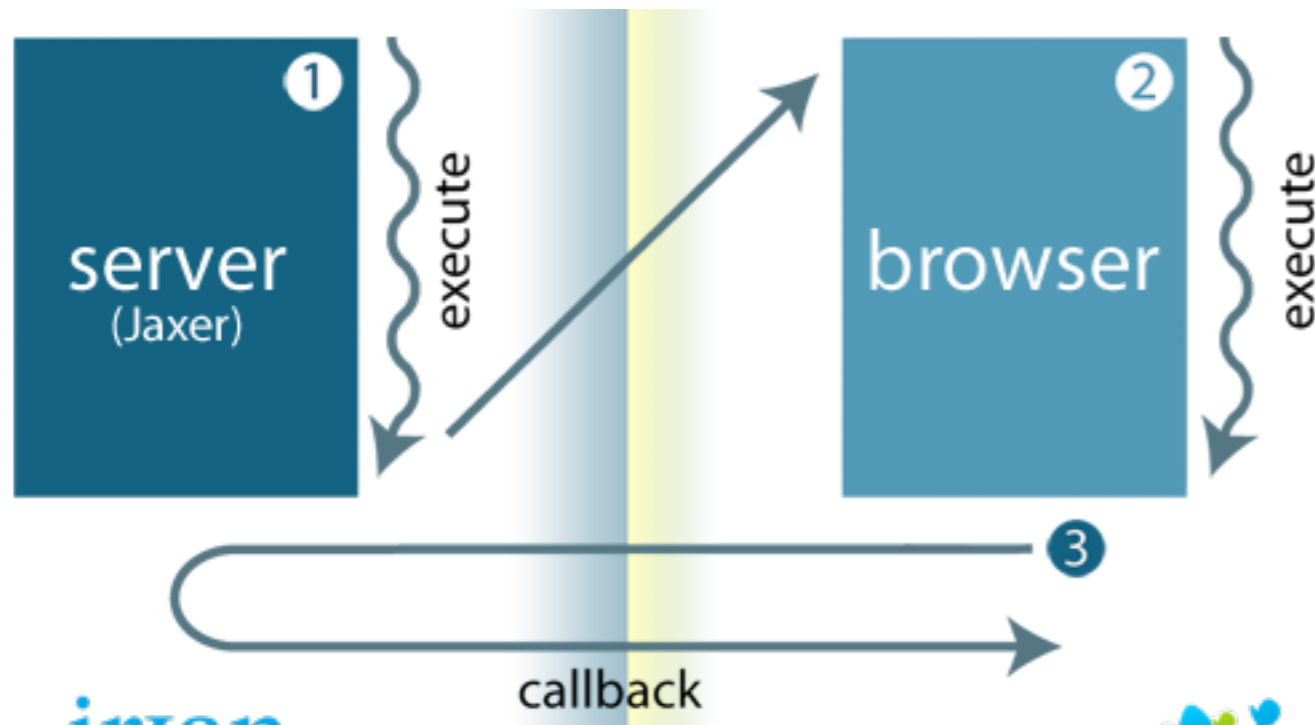
JS-Environments

aptana **Jaxer** The world's first Ajax server

- Use your Ajax, HTML, JavaScript and DOM skills server-side
- Integrate with databases, file systems, networks and more
- Just tag your JavaScript code to run on the server, the client, or both
- Easily deploy your Jaxer apps to Aptana Cloud from within Studio

Download Now

Free, open source software



- 1 Page executes on server and resulting page is sent to browser
- 2 Browser executes resulting page
- 3 Browser calls server asynchronously for new information

JS-Environments

- JavaScript on the server
 - Whole application is written in JavaScript
 - Multipass JavaScript execution
 - Server processes the JS first
 - Then it's sent to the client
 - Developer can decide which script blocks run on the server, which on client, or even both
 - <http://www.apptana.com/jaxer>

JS-Environments

```
<script runat="server">
  var resultSet = Jaxer.DB.execute("SELECT * FROM myTable");
  var newPrice = resultSet.rows[0].price;
</script>
```

```
<script runat="both">
  function validateCreditCard(number) {
    if (!...) throw new Error("...");
  }
</script>
```

```
<script runat="client">
  function uptime() {
    $('#ajaxSpinner').show();
    runUptime.async(updateProcessOutput);
  }
  function updateProcessOutput(res) {
    $('#processOutput').append(res);
    $('#ajaxSpinner').hide();
  }
</script>
```

JS-Environments



Get Adobe AIR

Download now >

Download applications
built for Adobe AIR >



Download Titanium Preview Release 3 (PR 3)



JS-Environments

- JavaScript on the desktop
 - Runtime environments for everything that would run in a browser
 - Enhanced JS APIs
 - File-Access
 - OS integration
 - Notifications
 - Statusbar
 - UI
 - Adobe Air
 - <http://www.adobe.com/products/air/>
 - Titanium
 - <http://titaniumapp.com/>

JS-Environments

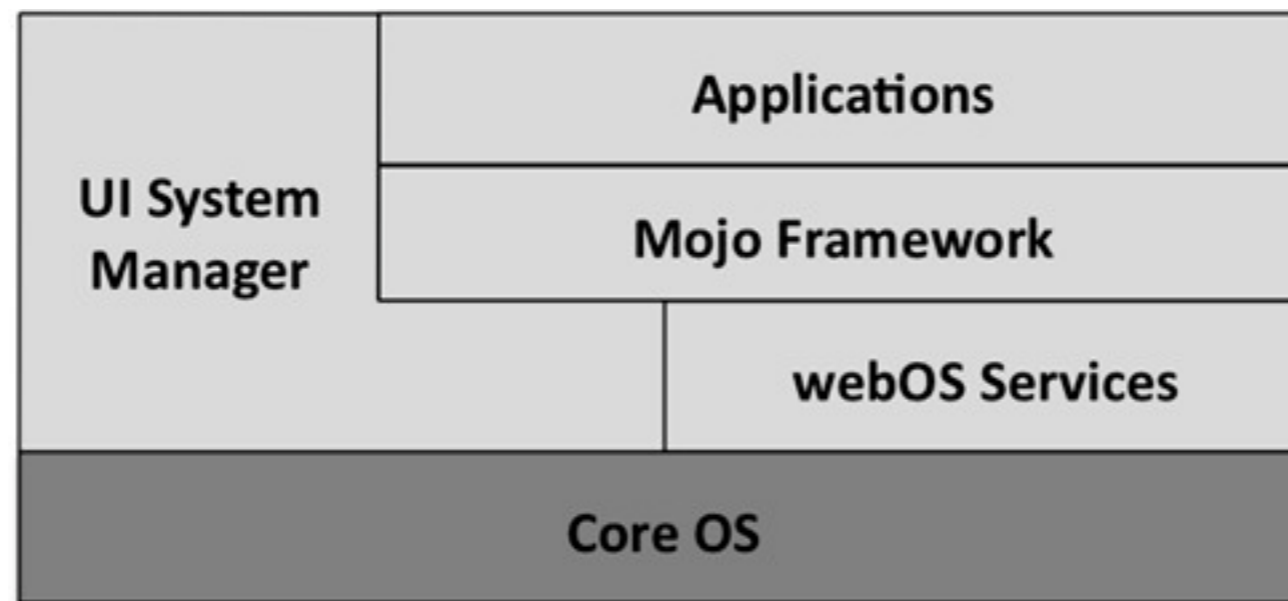


JS-Environments

- JavaScript on the phone

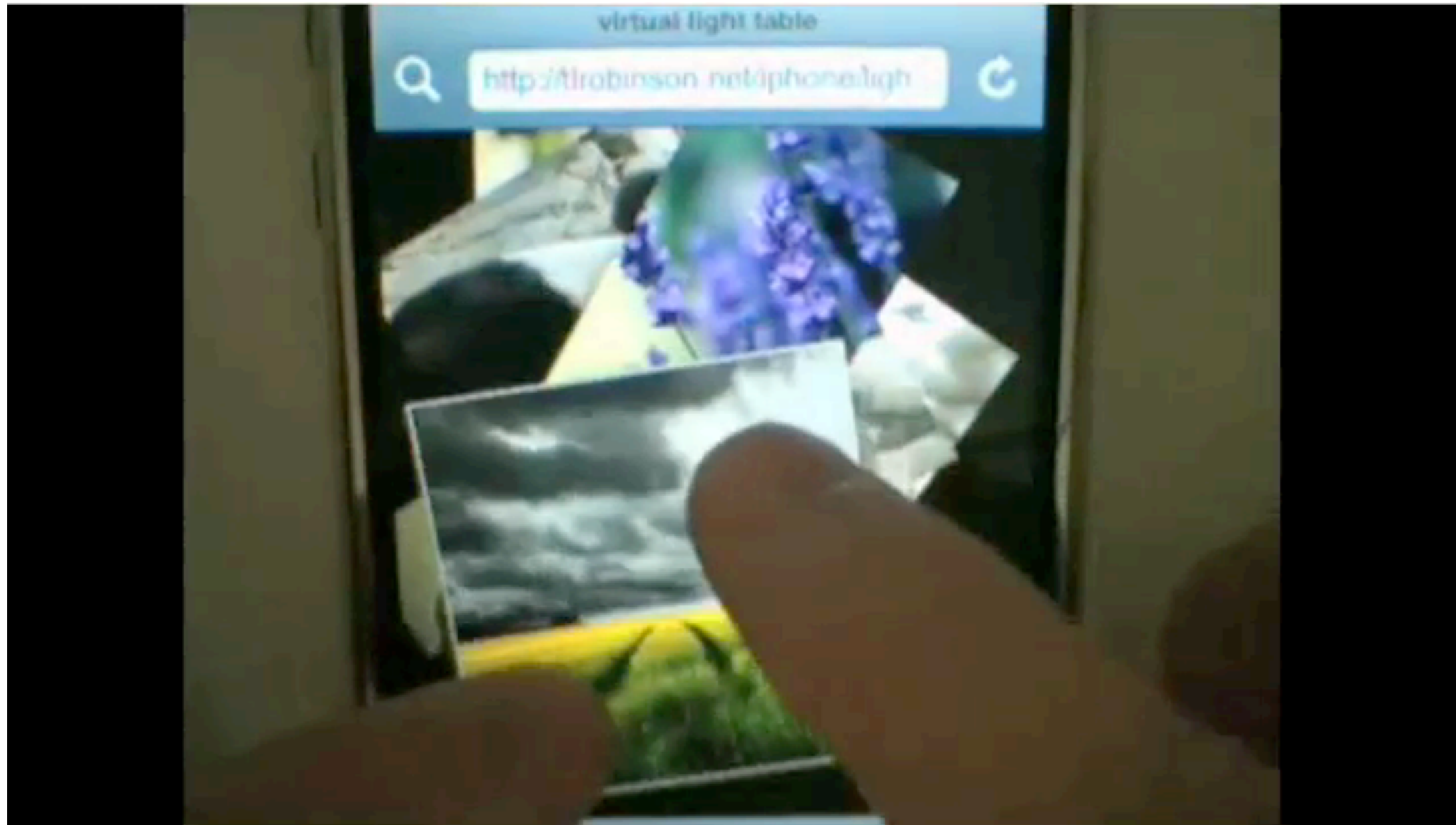
- Palm Pre features webOS and the Mojo Application Framework

- “Palm webOS applications are easy to write using Mojo, a new application framework based on the HTML5, CSS, and JavaScript standards that web developers already know and love.”



<http://developer.palm.com/>

JS Environments



Summary

- Lots of enhancements to browser functionality
- New JS APIs which make JS more powerful
- Better JS interpreters which make more complex applications possible
- ~~Browser~~ becomes an application platform

HTML/CSS/JaVaScRipt

- Already now very powerful combinations are possible
 - Number crunching (workers) graphically complex (canvas, css) applications, dealing with huge amounts of data (storage)

The future

- A lot of the things planned for this section already have been shown in “Here and now”
- But there is still some more
 - HTML5
 - JS
 - Development Tools
 - ...

HTML5

- Besides ...
 - Canvas
 - Storage
 - Workers
 - Audio/Video
- ... there are even more nice additions...

HTML5

- New form elements
 - Date & Time
 - URL / E-mail
 - Color
 - ...
- New section elements
 - section
 - article
 - ...
- New misc. elements
 - meter
 - progress
 - ...
- Web fonts

JS.next

- The JavaScript scripting language got standardized under the name ECMAScript
- The last update to ECMAScript (3rd edition) was in 1999
- ECMAScript 4 draft was around for quite a while
- ActionScript (Adobes Flash/Flex scripting language) is heavily influenced by ECMAScript 4
- But: ECMAScript 4 as it was drafted got abandoned

ECMAScript 4

- ES4 was planned as a major update to ES3
 - Introduced namespaces
 - Introduced optional typing
 - Introduced real classes
- ES4 really would have made JS another language
- There have been conflicts within the standards committee wether or not this was the right way to go
- Then there was Harmony

Harmony

The Ecma TC39 **meeting** in **Oslo** at the end of July was very productive, and if we keep working together, it will be seen as seminal when we look back in a couple of years. Before this meeting, I worked with John Neumann, TC39 chair, and ES3.1 and ES4 principals, especially Lars Hansen (Adobe), Mark Miller (Google), and Allen Wirfs-Brock (Microsoft), to unify the committee around shared values and a common roadmap. This message is my attempt to announce the main **result** of the meeting, which I've labeled "**Harmony**".

<https://mail.mozilla.org/pipermail/es-discuss/2008-August/006837.html>

Harmony

1. **Focus** work on **ES3.1** with full collaboration of all parties, and target two interoperable implementations by early next year.
2. Collaborate on the **next step beyond ES3.1**, which will include syntactic extensions but which will be **more modest** than ES4 in both semantic and syntactic innovation.
3. Some **ES4 proposals** have been deemed **unsound for the Web**, and are off the table for good: packages, namespaces and early binding. This conclusion is key to Harmony.
4. **Other goals** and ideas **from ES4** are being **rephrased** to keep consensus in the committee; these include a notion of classes based on existing ES3 concepts combined with proposed ES3.1 extensions.

<https://mail.mozilla.org/pipermail/es-discuss/2008-August/006837.html>

Development Tools

- Finally also the development tools are getting better (in every browser)

The screenshot shows a web browser window with the address bar displaying `http://dojotoolkit.org/`. The page is the Dojo Toolkit homepage, featuring the Dojo logo and the tagline "great experiences ...for everyone". Navigation links include "demos", "downloads", "get started", "documentation", "support", "community", and "full menu". A search bar is located in the top right corner. Below the navigation bar, there are three prominent buttons: "see it in action" (with a magnifying glass icon), "download now!" (with a download icon), and "documentation" (with a notepad icon). A banner at the bottom of the main content area reads: "Solve problems faster. Create better user experiences. Liberally licensed. Everything you need, all in one place."

Below the banner, there are three sections: "core" (Small, fast, deep.), "dijit" (Great interface widgets), and "dojoX" (The future, today.).

At the bottom of the browser window, there is a code editor with a debugger. The code editor shows a file named `common.js` with the following code:

```
// for now, no fx love but maybe in a bit.
dojo.query("#nav-fullmenu").onclick(function(e){
  var m=dojo.byId("main-menu-block-top");
  var n=e.target.parentNode;
  if(m){
    if(!d_fullMenuState){
      dojo.removeClass(n, "closed");
      dojo.addClass(n, "opened");
      dojo.fx.wipeIn({
        node: m,
        duration:400,
        onEnd: function(){ d_fullMenuState=!d_fullMenuState; }
      }).play();
    } else {
      dojo.removeClass(n, "opened");
      dojo.addClass(n, "closed");
      dojo.fx.wipeOut({
        node: m,
        duration:400,
        onEnd: function(){ d_fullMenuState=!d_fullMenuState; }
      }).play();
    }
  }
});
```

The debugger shows the following call stack:

- (anonymous function) common.js:33

The scope variables section shows the following variables:

- Local**
 - `e`: MouseEvent
 - `m`: undefined
 - `n`: undefined
 - `this`: HTMLLIElement
- Closure**
 - `b`: HTMLDivElement
 - `m`: HTMLDivElement
 - `n`: HTMLDivElement
 - `s`: CSSStyleDeclaration
- Global**: DOMWindow

The status bar at the bottom of the code editor shows 3 errors.

The screenshot shows a web browser window displaying the Dojo Toolkit homepage. The browser's address bar shows the URL `http://dojotoolkit.org/`. The homepage features the Dojo logo and the tagline "great experiences ...for everyone". A navigation bar includes links for "demos", "downloads", "get started", "documentation", "support", "community", and a "full menu" button. A sidebar on the right contains links for "login" and "register". The main content area has three colored boxes: a green box with a magnifying glass icon and the text "see it in action", a yellow box with a download icon and the text "download now!", and an orange box with a notepad icon and the text "documentation". Below these boxes is a dark banner with the text "Solve problems faster. Create better user experiences. Liberally licensed. Everything you need, all in one place."

Overlaid on the bottom of the browser window is a JavaScript debugger window. The "Script" tab is active, showing the source code of the Dojo Toolkit. The code is as follows:

```

25     if(b){
26         dojo.addClass(dojo.byId("nav-fullmenu"), "closed");
27         b.parentNode.insertBefore(n, b);
28     }
29
30     // hook up the full menu button
31     // for now, no fx love but maybe in a bit.
32     dojo.query("#nav-fullmenu").onclick(function(e){
33         var m=dojo.byId("main-menu-block-top");
34         var n=e.target.parentNode;
35         if(m){
36             if(!d_fullMenuState){
37                 dojo.removeClass(n, "closed");
38                 dojo.addClass(n, "opened");
39                 dojo.fx.wipeIn({
40                     node: m,
41                     duration:400,
42                     onEnd: function(){ d_fullMenuState=!d_fullMenuState; }
43                 });
44             }
45         }
46     });

```

The "Watch" tab is also active, showing the following watch expressions:

- this**: `li#nav-fullmenu.closed`
- scopeChain**: `[Call e=Event click, Call b=div#banner-block.banner-home, Window dojotoolkit.org 0=Call 1=Call 2=window]`
- e**: `click clientX=960, clientY=69`
- m**: `undefined`
- n**: `undefined`

The screenshot shows a Windows Internet Explorer browser window displaying the Dojo Toolkit website. The browser's address bar shows the URL `http://dojotoolkit.org/`. The website features the Dojo logo and the tagline "great experiences ...for everyone". A navigation bar includes links for "demos", "downloads", "get started", "documentation", "support", "community", and "full menu". A large banner on the right side of the page has three buttons: "see it in action", "download now!", and "documentation".

Overlaid on the bottom right of the browser window is a developer tool window titled "Home | The Dojo Toolkit - Entwicklertools". The "Script" tab is active, showing a JavaScript file named `common.js`. The code in the editor includes a function `if(b)` that manipulates the DOM using Dojo's `dojo.addClass`, `dojo.removeClass`, and `dojo.fx.wipeIn` methods. The code is as follows:

```

25  if(b){
26      dojo.addClass(dojo.byId("nav-fullmenu"), "closed");
27      b.parentNode.insertBefore(n, b);
28  }
29
30  // hook up the full menu button
31  // for now, no fx love but maybe in a bit.
32  dojo.query("#nav-fullmenu").onclick(function(e){
33      var m=dojo.byId("main-menu-block-top");
34      var n=e.target.parentNode;
35      if(m){
36          if(!d_fullMenuState){
37              dojo.removeClass(n, "closed");
38              dojo.addClass(n, "opened");
39              dojo.fx.wipeIn({
40                  node: m,
41                  duration:400,
42                  onEnd: function(){ d_full
43              }).play();
44          } else {
45              dojo.removeClass(n, "opened");
46              dojo.addClass(n, "closed");
47              dojo.fx.wipeOut({
48                  node: m,
49                  duration:400,
50                  onEnd: function(){ d_full
51              }).play();
52          }
53      }

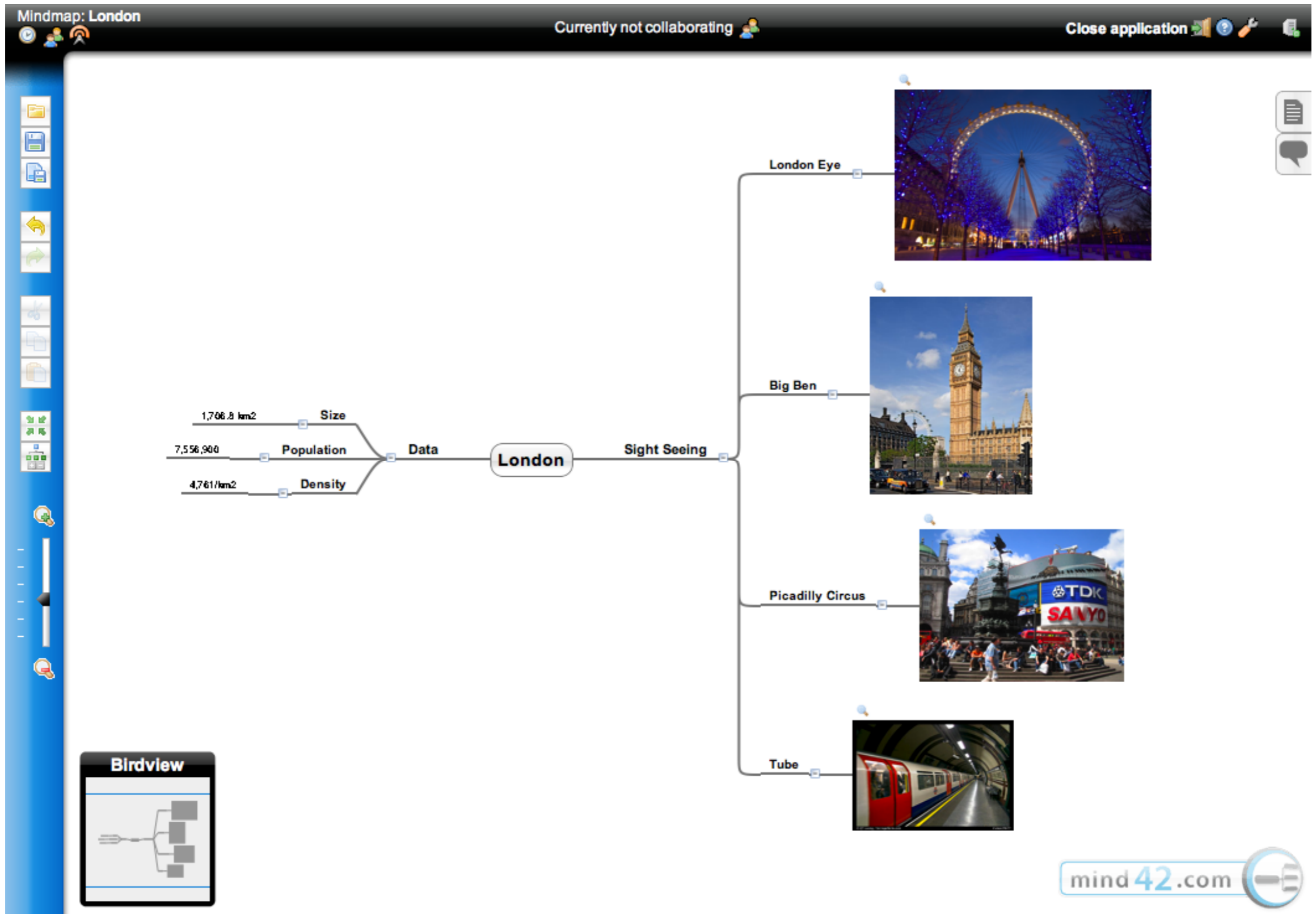
```

On the right side of the developer tool window, the "Lokale Variablen" (Local Variables) pane is open, displaying a table of variables and their values:

Name	Wert	Typ
[Methods]	{...}	DispCEventObj
altKey	false	Boolean
altLeft	false	Boolean
behaviorCookie	0	Number
behaviorPart	0	Number
bookmarks	null	HTMLBookmarkCollection
boundElements	{...}	DispHTMLElementCollection
button	0	Number
cancelBubble	false	Boolean
clientX	936	Number
clientY	69	Number
constructor	{...}	Object
contentOverflow	false	Boolean
ctrlKey	false	Boolean
ctrlLeft	false	Boolean
data	""	String
dataFld	""	String
dataTransfer	null	HTMLDataTransfer
fromElement	null	HTMLElement
keyCode	0	Number
nextPage	""	String
offsetX	524	Number
offsetY	8	Number
origin	""	String
propertyName	""	String

Inside Irian

- We're working with bleeding edge JS/CSS/HTML techniques for years now
 - Running service: Mind42 (<http://www.mind42.com>)
 - New platform: Spaaze (<http://www.spaaze.com>)
- Mind42
 - JS-SPA (JavaScript Single Page Application)
 - Rendering is done client side
 - Uses CSS/HTML/Canvas for drawing



Spaaze.com

- Spaaze.com will be our new service platform and playground for these kind of applications
 - MindSpaaze (former Mind42)
 - TodoSpaaze (Todo management application)
 - ProjectSpaaze (Project management application)
- Don't even try to navigate to spaaze.com yet
 - Just starting right now
 - Except: todo.spaaze.com



Thank you for your attention