



Java is a trademark of Sun Microsystems, Inc.



JavaOneSM

State of the Open Web

Mark Pilgrim
Patrick Chanezon
Google, Inc.

What is the open web?

What is the open web?

“The open web is a beautiful soup of barely compatible clients and servers. It comprises billions of pages, millions of users, and thousands of browser-based applications. You can access the open web with open source and proprietary browsers, on open source and proprietary operating systems, on open source and proprietary hardware.”

Again without the elevator pitch...

Again without the elevator pitch...

> HTML

Again without the elevator pitch...

- > HTML
- > CSS

Again without the elevator pitch...

- > HTML
- > CSS
- > JavaScript (DOM)

Again without the elevator pitch...

- > HTML
- > CSS
- > JavaScript (DOM)
- > SVG, MathML, audio, video...

Who cares about the open web?

Who cares about the open web?

- > Google

Who cares about the open web?

- > Google
- > probably some other people, too

Why does Google care?

Why does Google care?

- > Nobody “owns” the open web

Why does Google care?

- > Nobody “owns” the open web
- > It works on all platforms

Why does Google care?

- > Nobody “owns” the open web
- > It works on all platforms
- > You can deploy web applications without asking anyone's permission

Why does Google care?

- > Nobody “owns” the open web
- > It works on all platforms
- > You can deploy web applications without asking anyone's permission
- > You can build your own browser without asking anyone's permission

Why does Google care?

- > Nobody “owns” the open web
- > It works on all platforms
- > You can deploy web applications without asking anyone's permission
- > You can build your own browser without asking anyone's permission
- > You can index it without asking anyone's permission

HTML 5



HTML 5

- > Evolution of HTML 4
- > Started outside W3C
- > “Web Forms 2.0”
- > “Web Applications 1.0”
- > Folded back into W3C
- > “HTML 5”



HTML 5

> <canvas>

- 2-D drawing surface
- Fully scriptable



HTML 5

> <canvas>

- 2-D drawing surface
- Fully scriptable

```
ctx = a_canvas_element.getContext('2d');  
ctx.fillStyle = "rgb(200,0,0)";  
ctx.fillRect(10, 10, 55, 50);
```



HTML 5

> <video>

- Like , but moving
- Native UI for controls
- Fully scriptable (start, stop, pause, rewind, CC)
- Apply CSS transformations to video in real time
- Apply SVG filters to video in real time



HTML 5

> <video>



```
<video src=mymovie.ogg>
```

HTML 5

> <video>



```
<video src=mymovie.ogg  
controls>
```


HTML 5

> <video>

```
<video src=mymovie.ogg  
      controls  
      autoplay>
```



HTML 5

> <video>



```
<video src=mymovie.ogg  
      controls  
      autoplay  
      loop>
```

HTML 5

> <video>



```
<video src=mymovie.ogg  
      controls  
      autoplay  
      loop  
      poster=frame.jpg>
```

HTML 5

> <audio>

- Like <video> without the video
- Native UI for controls
- Fully scriptable (start, stop, pause, rewind)



HTML 5

> `<audio>`



`<audio src=my-podcast.ogg>`

HTML 5

> `<audio>`



```
<audio src=my-podcast.ogg  
controls>
```

HTML 5

> `<audio>`

```
<audio src=my-podcast.ogg  
      controls  
      autoplay>
```



HTML 5

> <audio>



```
<audio src=my-podcast.ogg  
      controls  
      autoplay  
      loop>
```


HTML 5

- > Offline applications



HTML 5

- > Offline applications
 - Huh?



HTML 5

- > Offline applications
 - You heard me



HTML 5

- > Offline applications
 - Application manifests
 - Local storage
 - Background (threaded) JavaScript



HTML 5

- > Offline applications



```
<html manifest="/sitemanifest">
```

HTML 5

> Offline applications



CACHE MANIFEST

jsfile1.js

jsfile2.js

styles.css

/images/image1.png

/images/image2.png

HTML 5

> Offline applications



```
if (window.applicationCache.status == 0) {  
    // Loaded from network  
} else {  
    // Loaded from AppCache (offline)  
}
```

HTML 5

- > Local database storage



```
var database = openDatabase("Database  
Name", "Database Version");
```


HTML 5

- > Local database storage



```
var database = openDatabase("Database  
    Name", "Database Version");  
database.executeSql("SELECT * FROM test",  
    function(result1) {  
        // do something with the results  
    }  
}
```

HTML 5

- > Offline applications
 - Already here
 - iPhone
 - Gmail, Gcalendar for mobile

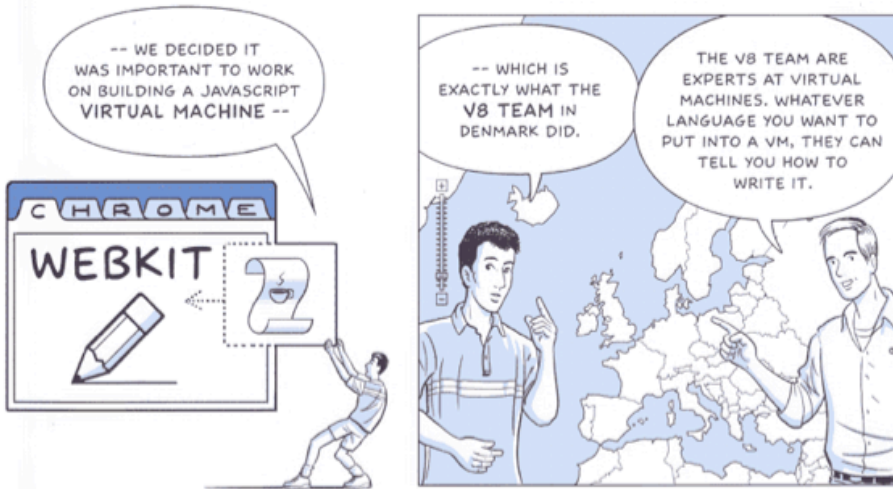
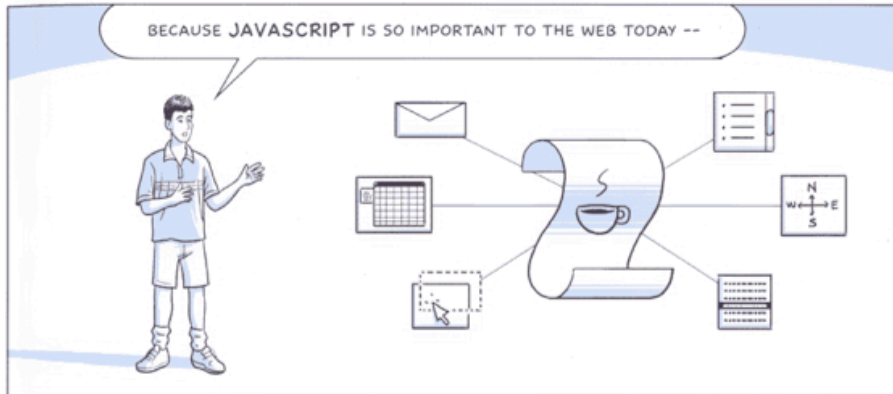


"gmail for mobile html5 series"

What's New in JavaScript

- > Fast JavaScript VMs (V8, Squirrelfish, etc.)
- > "programming in the large" support like ECMAScript 3.1/4
- > query-like syntaxes when working with the DOM
- > Demos
- > State of cross-browser support

Fast JavaScript VMs



The Challenge

- JavaScript is a highly dynamic language
- Objects are basically hash maps
- Properties are added and removed on-the-fly
- Prototype chains are modified during execution
- 'eval' can change the calling context
- 'with' introduces objects in the scope chain dynamically



V8 Design Decisions



Design Goals

- Make large object-oriented programs perform well
- Fast property access
- Fast function calls
- Fast and scalable memory management

Key V8 Components

- Hidden classes and class transitions
- Compilation to native code with inline caching
- Efficient generational garbage collection



V8 Internals



Hidden Classes

- Wanted to take advantage of optimization techniques from statically typed object oriented languages
- Introduced the concept of hidden classes to get there
- Hidden classes group objects that have the same structure

Hidden Classes by Example

- JavaScript objects constructed in the same way should get the same hidden class
- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**

Hidden Classes by Example

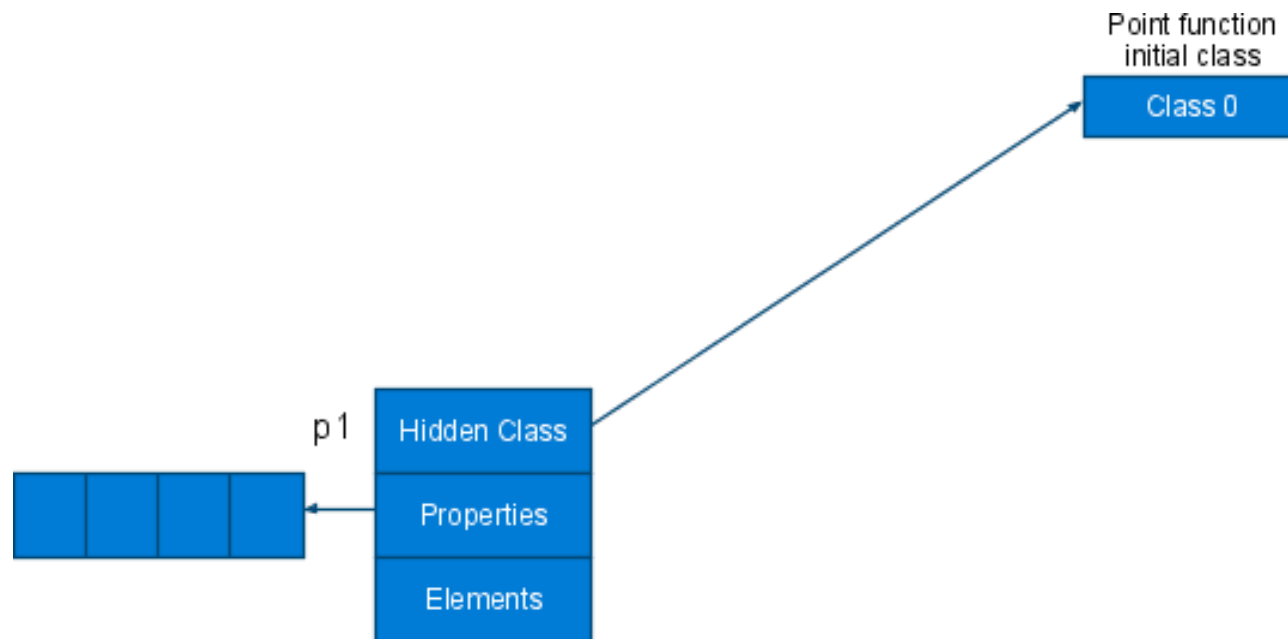
- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**

Point function
initial class

Class 0

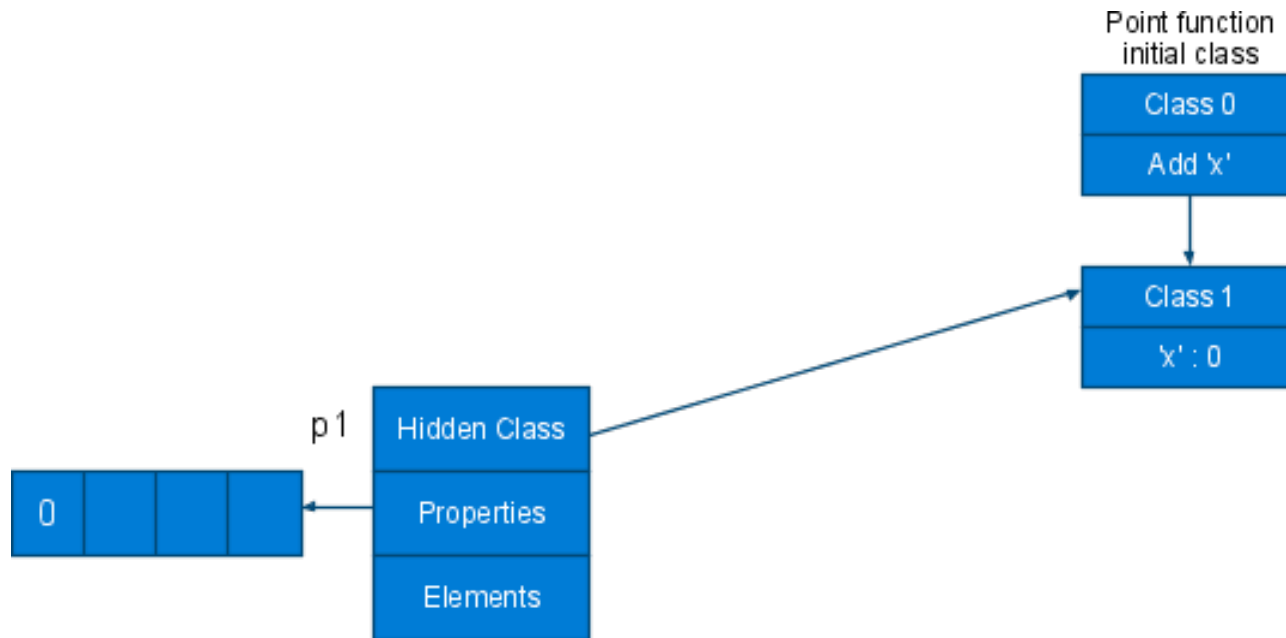
Hidden Classes by Example

- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



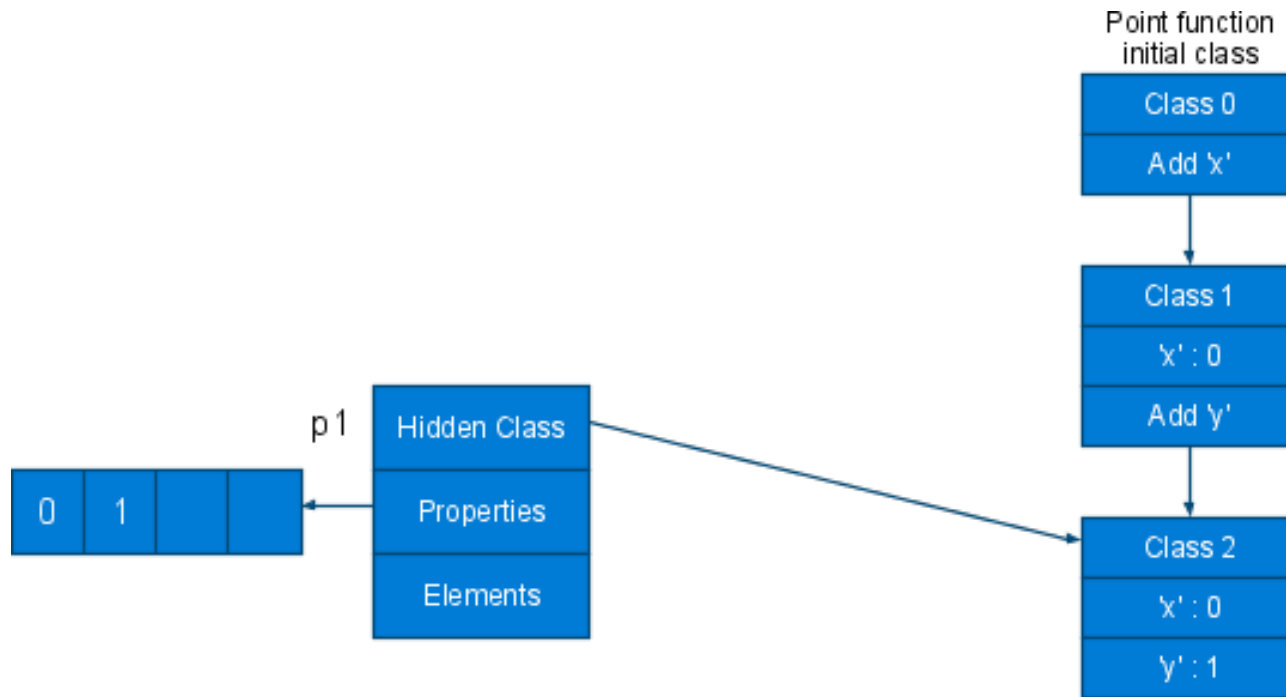
Hidden Classes by Example

- **function Point(x, y) {**
- **→ this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



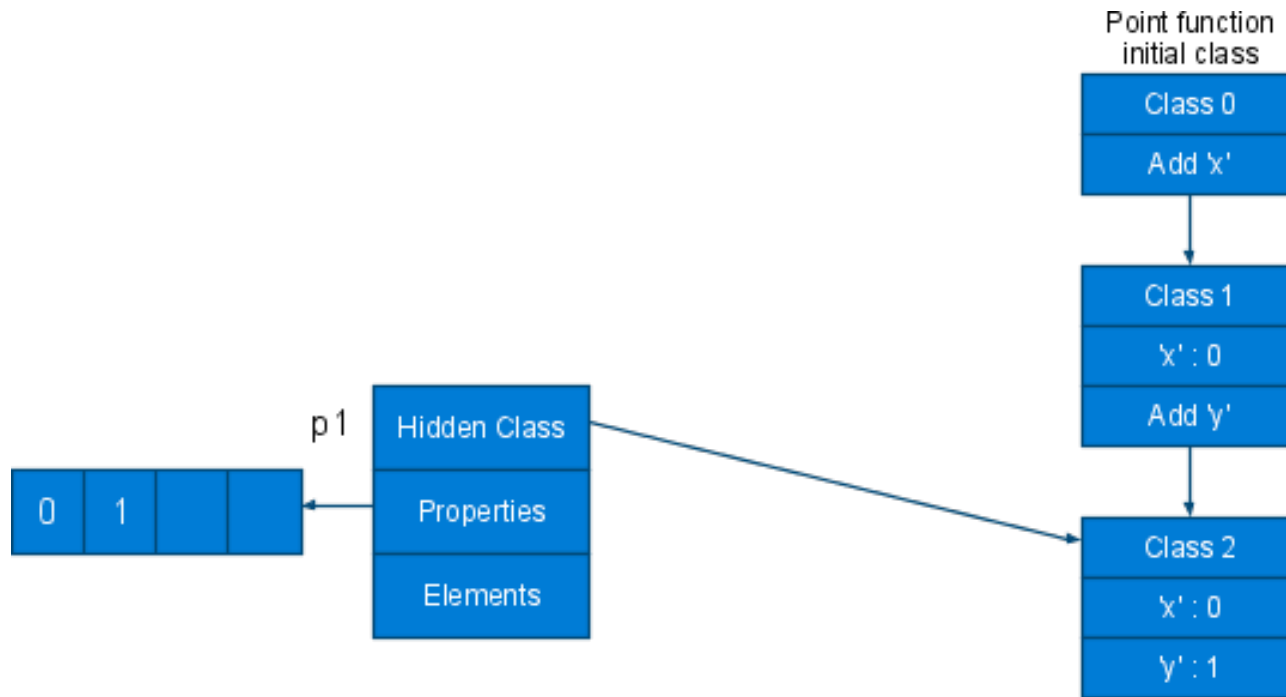
Hidden Classes by Example

- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



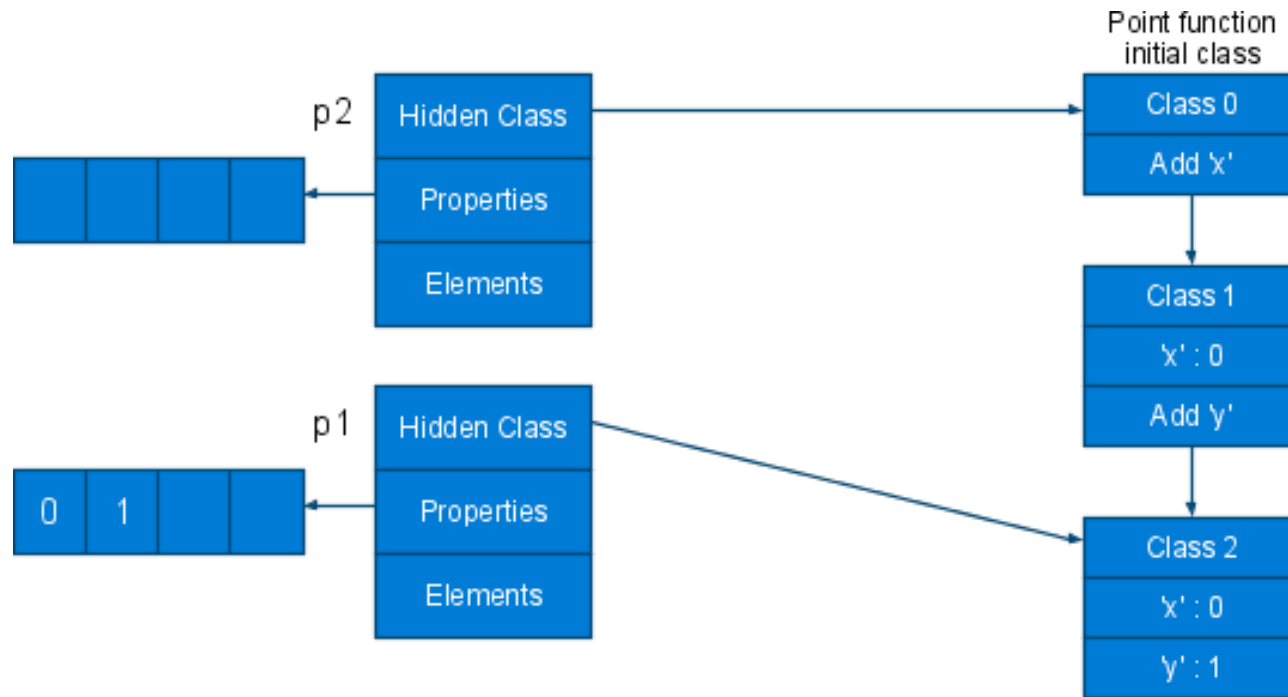
Hidden Classes by Example

- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



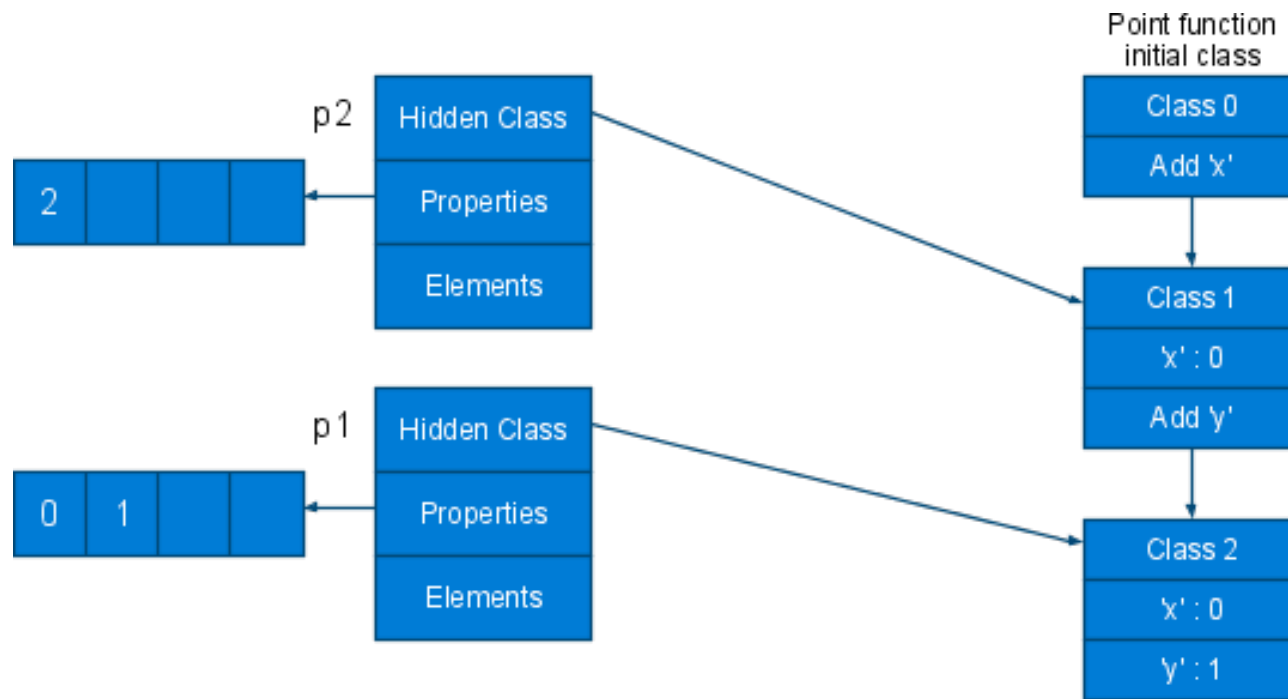
Hidden Classes by Example

- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



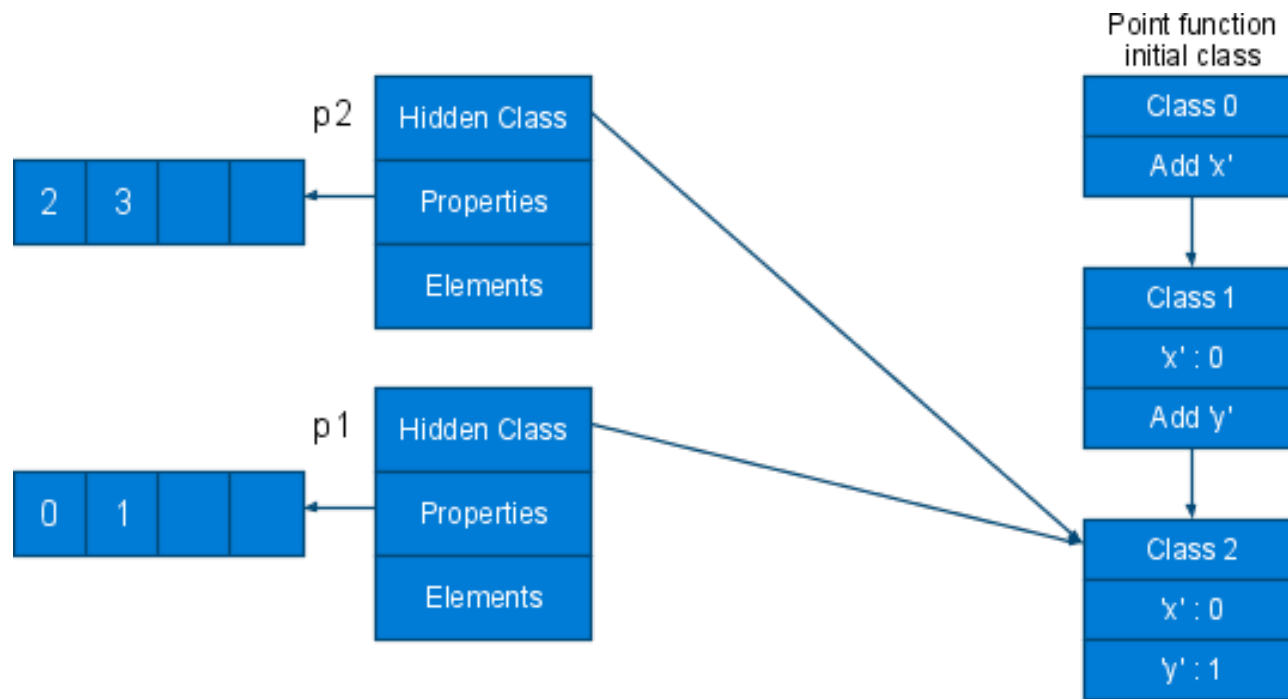
Hidden Classes by Example

- **function Point(x, y) {**
- **→ this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



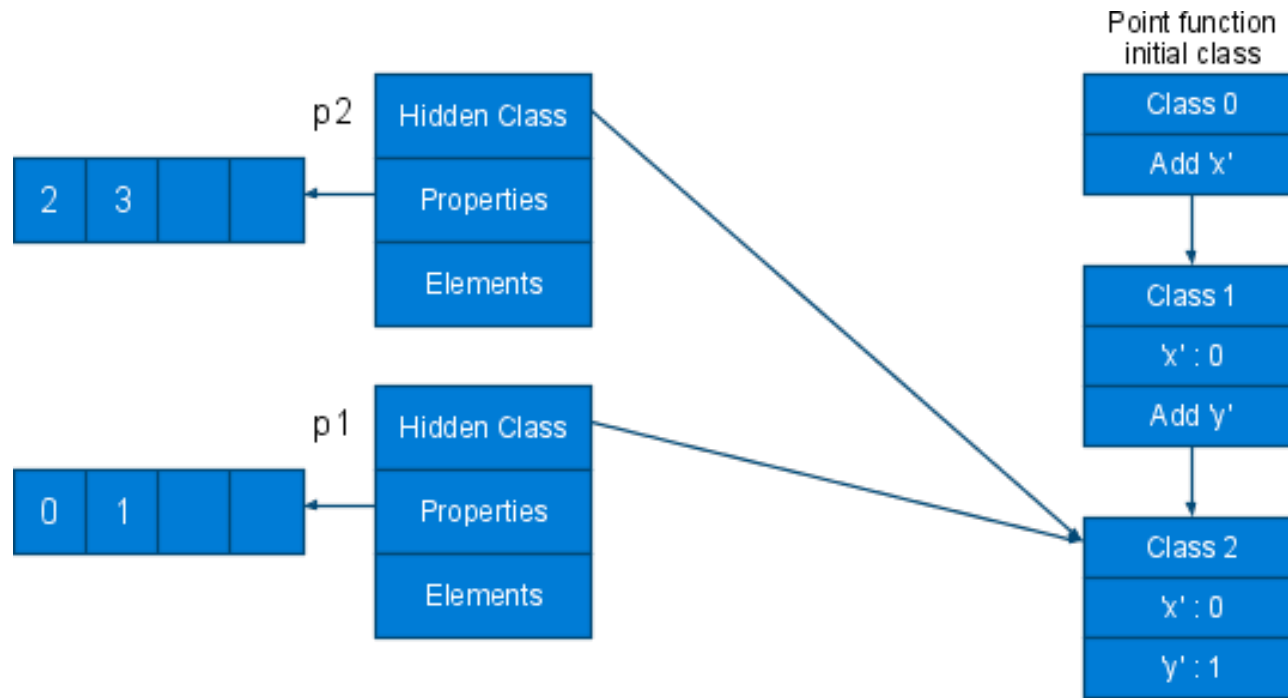
Hidden Classes by Example

- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



Hidden Classes by Example

- **function Point(x, y) {**
- **this.x = x;**
- **this.y = y;**
- **}**
- **var p1 = new Point(0,1);**
- **var p2 = new Point(2,3);**



How Dynamic is JavaScript?

- In 90% of the cases, only objects having the same map are seen at an access site
- Hidden classes provides enough structure to use optimization techniques from more static object-oriented language
- We do not know the hidden class of objects at compile time
- We use runtime code generation and a technique called inline caching

Inline Caching

- ...

- load 'x'



Full generic
lookup

- ...

- load 'y'



Full generic
lookup

- ...

Inline Caching

- ...

- load 'x'



- ...

- load 'y'

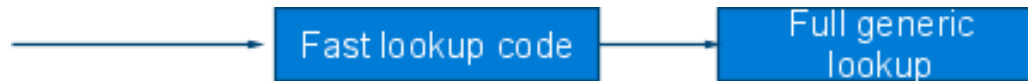


- ...

Inline Caching

- ...

- load 'x'



- ...

- load 'y'

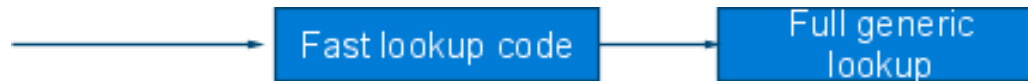


- ...

Inline Caching

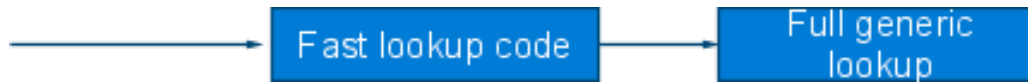
- ...

- load 'x'



- ...

- load 'y'



- ...

Efficient Memory Management

- Precise generational garbage collection
- Two generations
 - Young generation is one small, contiguous space that is collected often
 - Old generation is divided into a number of spaces that are only occasionally collected
 - Code space (executable)
 - Map space (hidden classes)
 - Large object space (>8K)
 - Old data space (no pointers)
 - Old pointer space
- Objects are allocated in the young generation and moved to the old generation if they survive in the young generation

Types of Garbage Collection

- Scavenge
 - Copying collection of only the young generation
 - Pause times $\sim 2\text{ms}$
- Full non-compacting collection
 - Mark-Sweep collection of both generations
 - Free memory gets added to free lists
 - Might cause fragmentation
 - Pause times $\sim 50\text{ms}$
- Full compacting collection
 - Mark-Sweep-Compact collection of both generations
 - Pause times $\sim 100\text{ms}$



Recent developments



Irregexp: New Regular Expression Engine

- V8 initially used a library from WebKit called JSCRE
- JSCRE did not fit well with the string types in V8 and did not perform well
- Implemented Irregexp giving a 10x speedup on regular expression matching on benchmark programs
- Irregexp implements full JS regular expressions and there is no fallback to other libraries

New Compiler Infrastructure

- Original compiler was very simple
- No static analysis of any kind
- No register allocation
- We have implemented a new compiler infrastructure which performs register allocation
- Still a one pass compiler
- Forms the basis for further native code optimizations



Object Heap Scalability



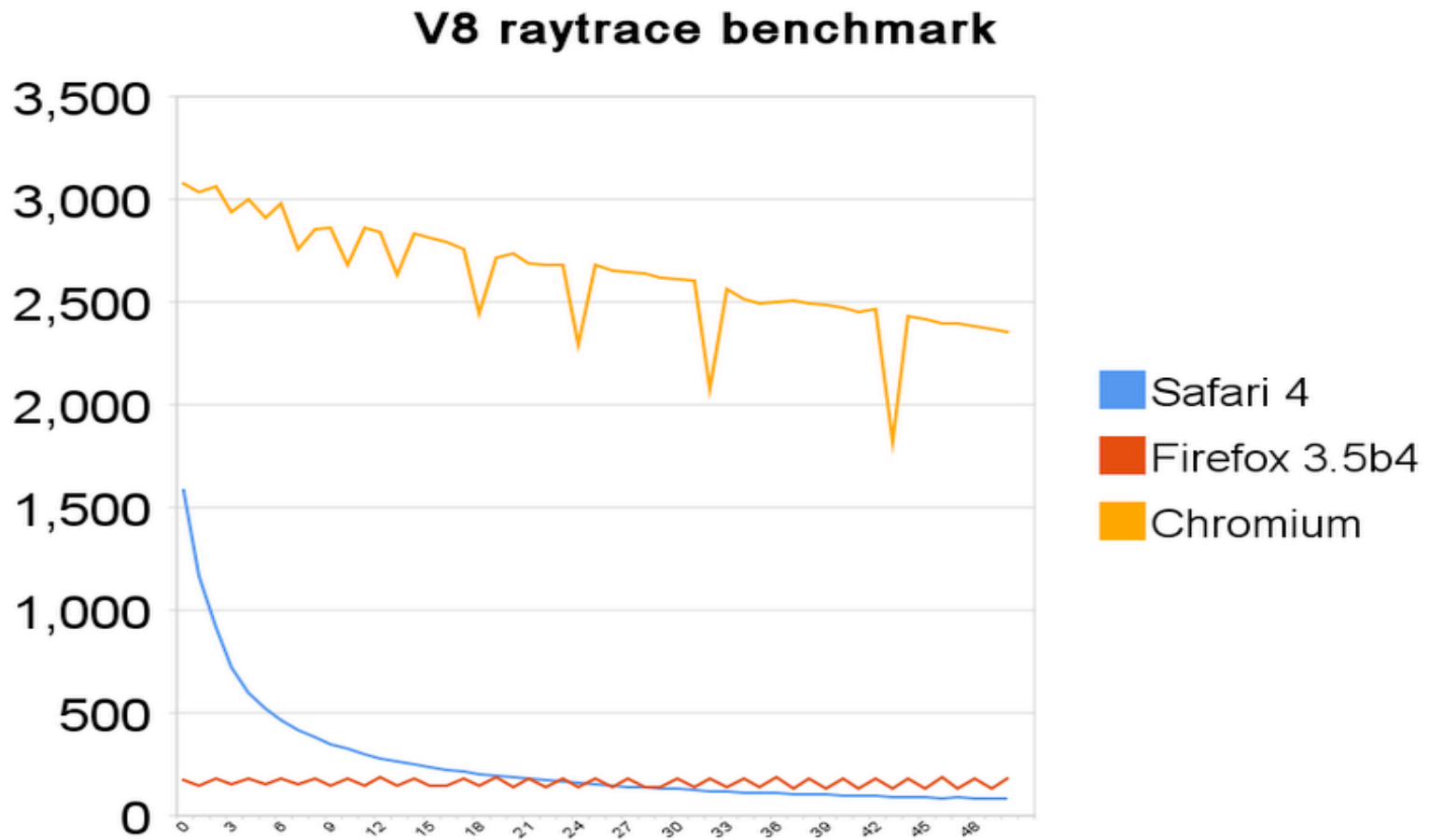
Scalability

- Users use multiple tabs running full applications
 - Mail
 - Calendar
 - News
- Applications are becoming bigger with more objects
- JavaScript execution should be fast in these situations
- The challenge is to scale well with respect to the size of the object heap
- The key to scaling well is generational garbage collection

Scalability Experiment

- Artificial scalability experiment approximating this situation
 - Raytrace benchmark from the V8 benchmark suite
 - Allocate extra live data on the side
 - 1MB of extra data per iteration

Scalability Experiment - Execution Speed



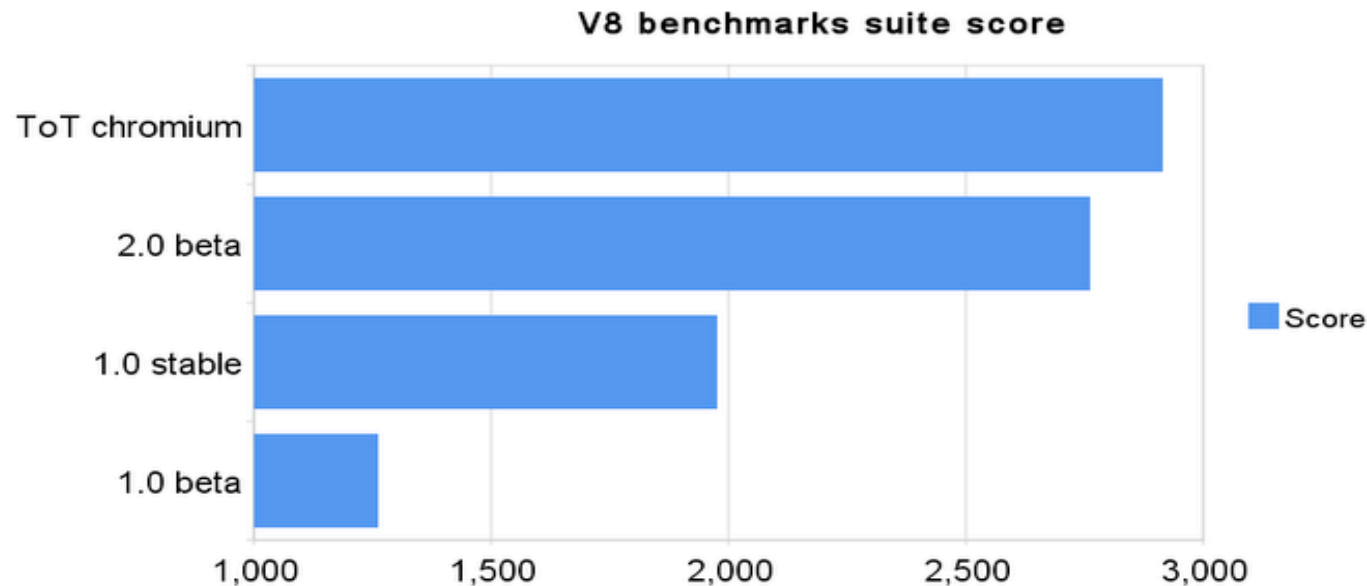
Bigger is better!

Scalability

- This experiment is artificial
- Try loading GMail in different browsers and then run JavaScript benchmarks in another tab
- Try out your own scalability experiments!

Summary

- V8 was designed for speed and scalability
- The goal is to raise the performance bar for JavaScript



- The full source code is available under a BSD license

Ecmascript versions smorgasbord

Edition 3: what you probably use today

Edition 4 (Historical): never completed, influenced version 5

Edition 5: final candidate draft of ECMA-262

completed and is available for public review and testing.

Edition 6 ("Harmony"): next-generation version of ECMAScript

JavaScript 1.6-1.9: code names for the interim versions of ECMAScript (Gecko) -> JavaScript 2.0 (ECMAScript 4)

"programming in the large" support like ECMAScript 3.1/4

Classes

Getters and Setters

Packages and Namespaces

Type annotations

Block scope let

Iterators and Generators

Extra array methods

Safer Json

Strict mode for easier debugging

Classes

```
class Programmer {  
    var name;  
    var city = 'Boston, MA';  
    const interest = 'computers';  
    function work() {}  
}
```

ECMAScript 5

Getters and Setters

```
Object.preventExtensions( obj )  
Object.isExtensible( obj )  
{  
    value: "test",  
    writable: true,  
    enumerable: true,  
    configurable: true  
}
```


ECMAScript 5

Getters and Setters

```
(function() {  
    var name = "John";  
  
    Object.defineProperty( obj, "name", {  
        get: function() { return name; },  
        set: function(value) { name = value; }  
    });  
})();
```

Packages and namespaces

```
namespace ns1
namespace ns4 = "www.ecma-international.org"
use default namespace ns
  package org.ecmascript.experiment {
    internal var v;
  }
package org.ecmascript.experiment {
  public function f(k) { v += k; return v-1 }
}
import org.ecmascript.experiment.*
f(37) // calls org.ecmascript.experiment.f
```

Type Annotations

```
type Point = { x: int, y: int }
```

Block scope with let

```
var x = 5;
```

```
var y = 0;
```

```
let (x = x+10, y = 12) {  
    print(x+y + "\n");  
}
```

```
print((x + y) + "\n");
```

Generators/Iterators

```
function fib() {  
    var i = 0, j = 1;  
    while (true) {  
        yield i;  
        var t = i;  
        i = j;  
        j += t;  
    }  
}  
  
var g = fib();  
for (var i = 0; i < 10; i++) {  
    document.write(g.next() + "<br>\n");  
}
```

Iterators, Array Comprehensions

```
for (var i in objectWithIterator)
{
    document.write(objectWithIterator[i] + "<br>\n");
}

function range(begin, end) {
    for (let i = begin; i < end; ++i) {
        yield i;
    }
}

var ten_squares = [i * i for each (i in range(0, 10))];
```

Expression closures

```
function(x) x * x
```

//instead of

```
function(x) { return x * x; }
```

Generator Expression

```
let it = (i + 3 for (i in someObj));  
try {  
    while (true) {  
        document.write(it.next() + "<br>\n");  
    }  
} catch (err if err instanceof  
StopIteration) {  
    document.write("End of record.<br>\n");  
}
```


Generator Expression

```
handleResults( i for ( i in obj ) if ( i >
3 ) );
```

```
function handleResults( results ) {
    for ( let i in results )
        // ...
}
```

ECMAScript 5

Extra array methods

indexOf, lastIndexOf, every, some, forEach, map, filter, reduce, and reduceRight.

Reflective meta-programming

ECMAScript 5

Safer Json

Use Crockford's json2.js

```
var obj = JSON.parse('{"name":"John"}');  
var str = JSON.stringify({ name: "John" });
```

Strict mode for easier debugging

`"use strict";`

like Perl strict mode, different than FF strict mode

deprecated methods from ES3 throw errors

defined variables, no evals, no messing with arguments,
no with

query-like syntaxes when working with the DOM

```
var cells =  
    document.querySelectorAll("#score>t  
body>tr>td:nth-of-type(2)");
```

"programming in the large" support like ECMAScript 3.1/4

Olav Junker Kjær's [Mascara](#)

<http://ecmascript4.com/>

ES4-> javascript translator in python

State of cross-browser support

- >IE 6-7 support JScript 5 (which is equivalent to ECMAScript 3, JavaScript 1.5)
- >IE 8 supports JScript 6 (which is equivalent to ECMAScript 3, JavaScript 1.5 - more bug fixes over JScript 5)
- >Firefox 1.0 supports JavaScript 1.5 (ECMAScript 3 equivalent)
- >Firefox 1.5 supports JavaScript 1.6 (1.5 + Array Extras + E4X + misc.)
- >Firefox 2.0 supports JavaScript 1.7 (1.6 + Generator + Iterators + let + misc.)
- >Firefox 3.0 supports JavaScript 1.8 (1.7 + Generator Expressions + Expression Closures + misc.)
- >The next version of Firefox will support JavaScript 1.9 (1.8 + To be determined)
- >Opera supports a language that is equivalent to ECMAScript 3 + Getters and Setters + misc.
- >Safari supports a language that is equivalent to ECMAScript 3 + Getters and Setters + misc.

Source John Resig <http://ejohn.org/blog/versions-of-javascript/>

Acknowledgements

Mads Sig Ager, Google for V8 slides

Brendan Eich

[http://www.sdtimes.com/blog/post/2009/04/16/
Brendan-Eich-Explains-ECMAScript-Fifth-Edition-
To-You.aspx](http://www.sdtimes.com/blog/post/2009/04/16/Brendan-Eich-Explains-ECMAScript-Fifth-Edition-To-You.aspx)

John Resig

<http://ejohn.org/blog/tags/ecmascript/>

What's new in CSS?

What's new in CSS?

Generated content

What's new in CSS?

Generated content

```
h1:before {content: "Chapter " counter(h1) ". "}
```

What's new in CSS?

Generated content

```
h1:before {content: "Chapter " counter(h1) ". "}  
h1 {counter-reset: h2}  
h2:before {  
    counter-increment: h2;  
    content: counter(h1) "." counter(h2) ". ";  
}
```

What's new in CSS?

Generated content

```
<h1>Your First Python Program</h1>
```

```
<h2>Diving in</h2>
```

```
<h2>Declaring functions</h2>
```

```
<h2>Writing readable code</h2>
```

What's new in CSS?

Generated content

Chapter 1. Your First Python Program

1.1 Diving in

1.2 Declaring functions

1.3 Writing readable code

What's new in CSS?

Generated content

```
h1:before {content: "Chapter " counter(h1) ". "}  
h1 {counter-reset: h2}  
h2:before {  
    counter-increment: h2;  
    content: counter(h1) "." counter(h2) ". ";  
}
```

Safari, Firefox, Chrome, Opera... and Internet Explorer (8)

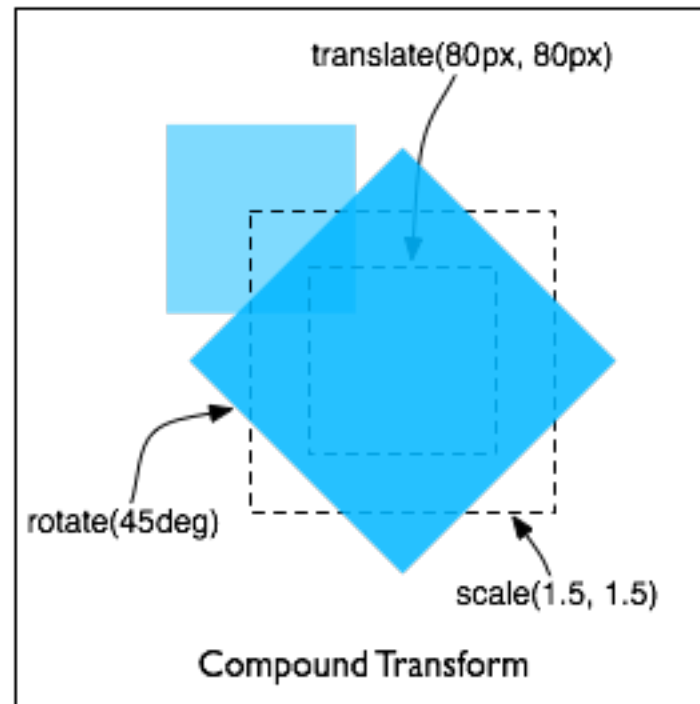
What's new in CSS?

Transforms

- `scale, scaleX, scaleY`
- `rotate`
- `translate, translateX, translateY`
- `skew, skewX, skewY`
- `matrix`

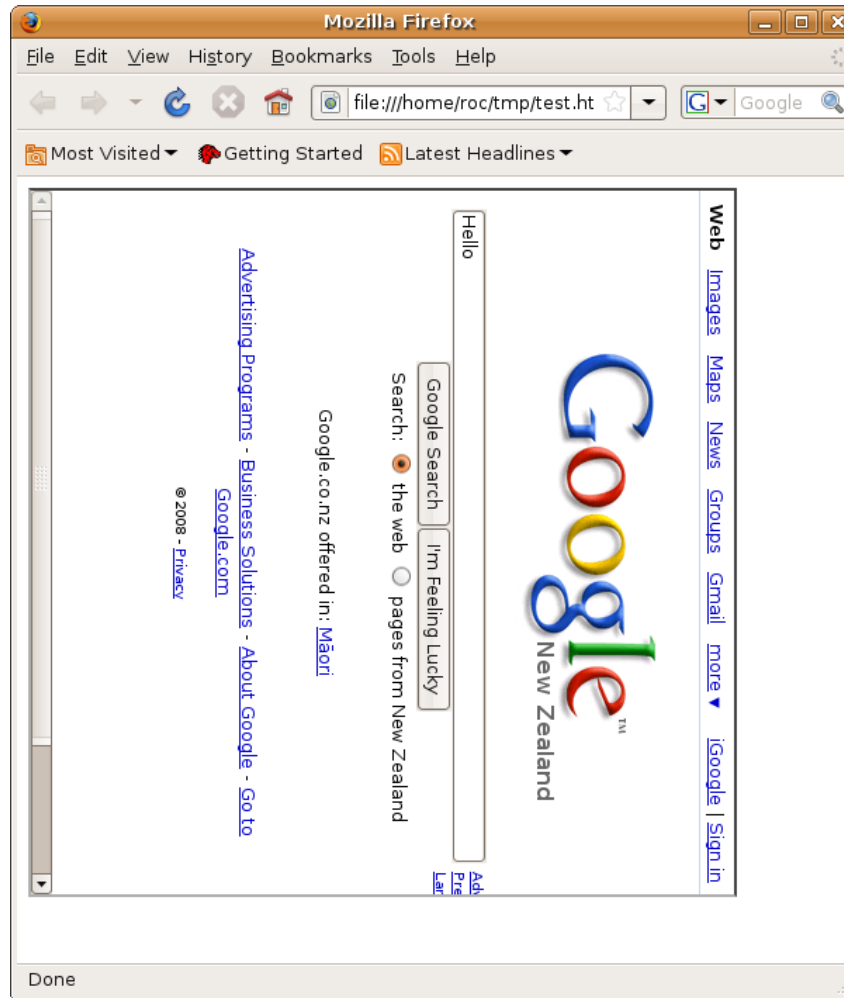
CSS Transforms

```
div {  
  height: 100px; width: 100px;  
  transform: translate(80px, 80px)  
             scale(1.5, 1.5) rotate(45deg);  
}
```



What's new in CSS?

Transforms



What's new in CSS?

Transforms

- `scale, scaleX, scaleY`
- `rotate`
- `translate, translateX, translateY`
- `skew, skewX, skewY`
- `matrix`

Safari 3, Firefox 3.5, Chrome,
iPhone, Android, Palm Pre, Blackberry Storm

CSS Animation

```
@keyframes 'wobble' {  
  0% {  
    left: 100px;  
  }  
  40% {  
    left: 150px;  
  }  
  60% {  
    left: 75px;  
  }  
  100% {  
    left: 100px;  
  }  
}
```

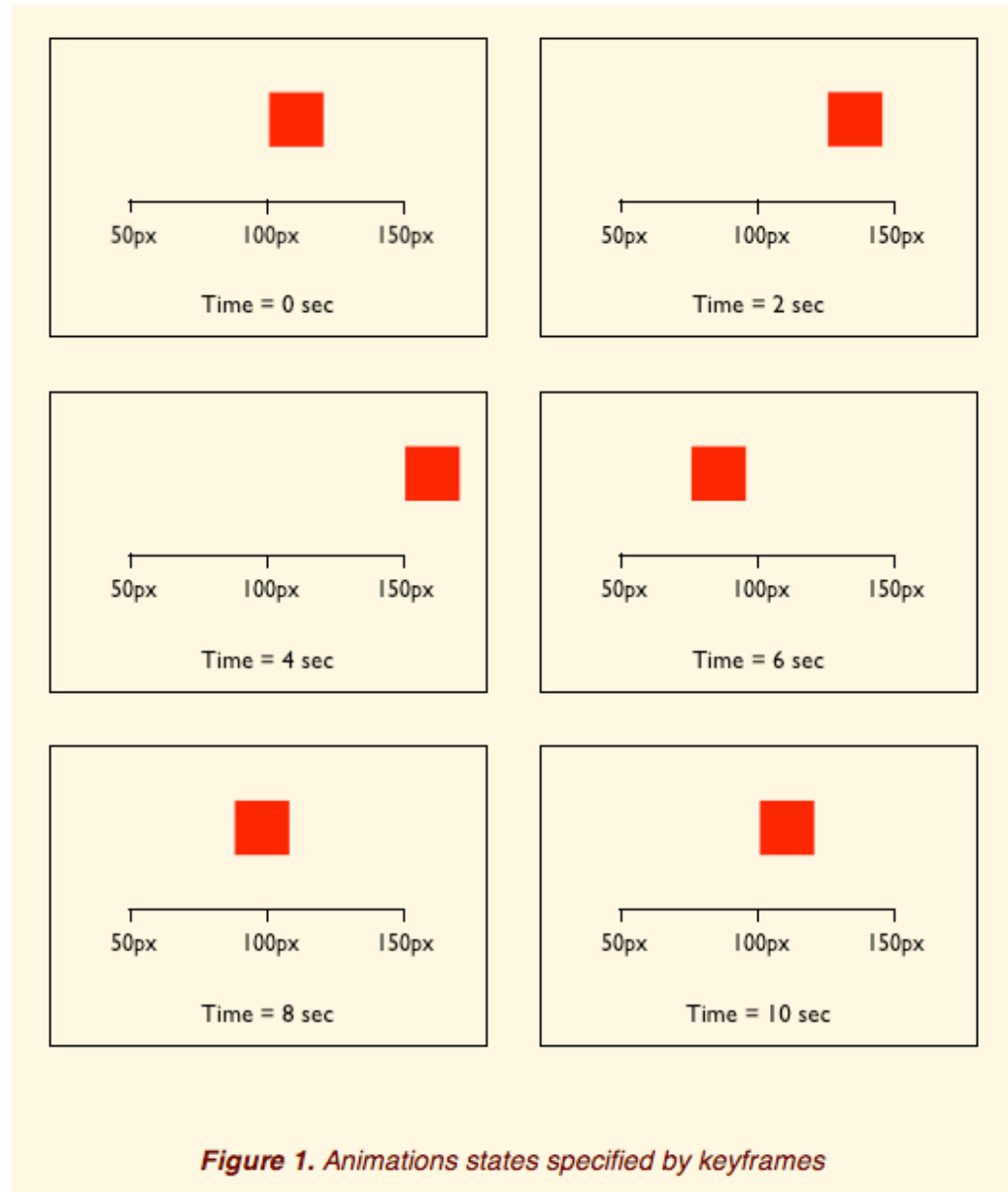


Figure 1. Animations states specified by keyframes

CSS Animation

```
div {  
    animation-name: 'wobble';  
    animation-duration: 5s;  
    animation-iteration-count: 10;  
}
```

What's new in CSS?

Embedded fonts

What's new in CSS?

Embedded fonts

... because the **web didn't look bad enough** already

What's new in CSS?

Embedded fonts

```
@font-face {  
    font-family: GoudyBookletter;  
    src: url (...);  
    format ("truetype");  
}
```

What's new in CSS?

Embedded fonts

```
@font-face {  
    font-family: GoudyBookletter;  
    src: url(goudybookletter.ttf);  
    format("truetype");  
}
```

Safari 3, Firefox 3.5, Chrome, iPhone, Android...

What's new in CSS?

Embedded fonts

```
@font-face {  
    font-family: GoudyBookletter;  
    src: url(goudybookletter.eot);  
}
```

Because Internet Explorer is... “special”

CSS Support in browsers

Selector	IE 5.5	IE 6	IE 7	IE8 as IE7	IE8 as IE8	FF 2	FF 3.0	FF 3.5b4	Saf 3.0 Win	Saf 3.1 Win	Saf 4.0b Win	Chrome 1	Chrome 2	Opera 9.62	Opera 10a	Konqueror 3.5-7
CSS 2	incorrect	incomplete	yes	yes	to be tested	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

CSS 2 has become the baseline of CSS support; without it a browser is decidedly backward.

CSS 3	minimal	incomplete	almost	to be tested	incomplete	almost	almost	almost	almost	almost	almost	almost	almost	almost	almost	almost
-----------------------	---------	------------	--------	--------------	------------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

The compatibility information here is about the CSS3 modules I test. It is not necessarily valid for the browsers' entire CSS3 support.

<http://www.quirksmode.org/compatibility.html>

What's new in vector graphics?

Canvas
SVG



13 years of web vector graphics history

- > 1996 Macromedia Flash 1
- > 1998 Macromedia and Microsoft VML
- > 1998 Adobe Systems and Sun Microsystems PGML
- > 1998 SVG W3C
- > 2001 SVG 1.0
- > 2003 SVG 1.1: modularization, Tiny and Basic
- > 2004 Apple Canvas -> WHATWG - HTML5, Mozilla
- > 2006 Microsoft Silverlight 1.0 (WPF/E)
- > 2007 Adobe Flash 9
- > 2008 SVG Tiny 1.2 (R) Full 1.2 (WD) core and modules

Web vector graphics today



99%



33%



20%

Microsoft®
Silverlight™

33%

HTML 5
Draft



4.8.11 The *canvas* element

OpenWeb vector graphics available in a browser near you

> Firefox, Safari (+iPhone), <http://a.deveria.com/caniuse/>, Opera, Chrome

> ~33% browsers natively

> Open Source JS Shims for Canvas and SVG (autumn 2009) support in IE

- Much of SVG 1.1 Full supported by all browsers except IE
- Most of Canvas (except for text extensions) supported by all browsers except IE

> No Flash in iPhone & Android

SVG (basic support) - Recommendation

Method of displaying basic Vector Graphics features using the embed or object elements

Resources: [Wikipedia](#) [Sample files](#)

	Internet Explorer	Firefox	Safari	Chrome	Opera
Far Past	6.0	2.0	3.1	0.2	9.0
Past	7.0				
Present	8.0	3.0	3.2	1.0	9.6
Near Future (2009)		3.5	4.0	2.0	10.0
Future (2010 or later)	9.0	4.0	4.*		10.*

Canvas (basic support) - Working Draft

Method of generating dynamic graphics using JavaScript

Resources: [Wikipedia](#) [Tutorial by Mozilla](#) [Animation kit experiment](#)

	Internet Explorer	Firefox	Safari	Chrome	Opera
Far Past	6.0	2.0	3.1	0.2	9.0
Past	7.0				
Present	8.0	3.0	3.2	1.0	9.6
Near Future (2009)		3.5	4.0	2.0	10.0
Future (2010 or later)	9.0	4.0	4.*		10.*

<canvas> tag and Context object

> Immediate mode graphics: fire and forget

```
<canvas id="clock" width="150" height="150">  
    
</canvas>
```

> In Javascript get a rendering context and draw

```
var canvas = document.getElementById("canvas"); if  
(canvas.getContext) { var ctx =  
canvas.getContext("2d"); ctx.fillStyle =  
"rgb(200,0,0)"; ctx.fillRect (10, 10, 55, 50); }
```

Demos

- > Yahoo Pipes
- > Bespin
- > All demos at <http://delicious.com/chanezon/j1+canvas>
- > More at <http://delicious.com/chanezon/canvas+demos>

Canvas Resources

- > <http://delicious.com/chanezon/canvas>
- > https://developer.mozilla.org/en/Canvas_tutorial
- > <http://dev.opera.com/articles/view/html-5-canvas-the-basics/>
- > <http://blog.mozbox.org/tag/demo>

SVG

- > XML vocabulary for Vector Graphics
- > Retained mode graphics
- > Versions: 1.0, 1.1, 1.2
- > Profiles: Full, Basic, Tiny
- > Safe to use: SVG 1.1 Full (except some features)
 - Not widely implemented: Fonts, Animation, some Filters

Coordinates, Viewbox

```
<?xml version="1.0" encoding="UTF-8"?><svg xmlns="http://www.w3.org/2000/svg"
  viewBox="50 50 500 400"    width="800px" height="600px"
id="ex1">  <g id="layer1">    <rect
style="fill:#fffc00;stroke:#fac305;stroke-width:20"
id="rect1" width="400" height="300"
  x="100" y="100" rx="20" ry="30"/>  </g></svg>
```




























Browser Support

> http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28SVG%29

			Gecko	WebKit
SVG	1.1	Tiny (SVGT)	Partial	Partial
		Basic (SVGB)	Partial	Partial
		Full	Partial	Partial
	1.2	Tiny	No	No

- > SVG 1.1 partially supported in Firefox, Safari (+iPhone), Opera, Chrome
- > 33% browsers

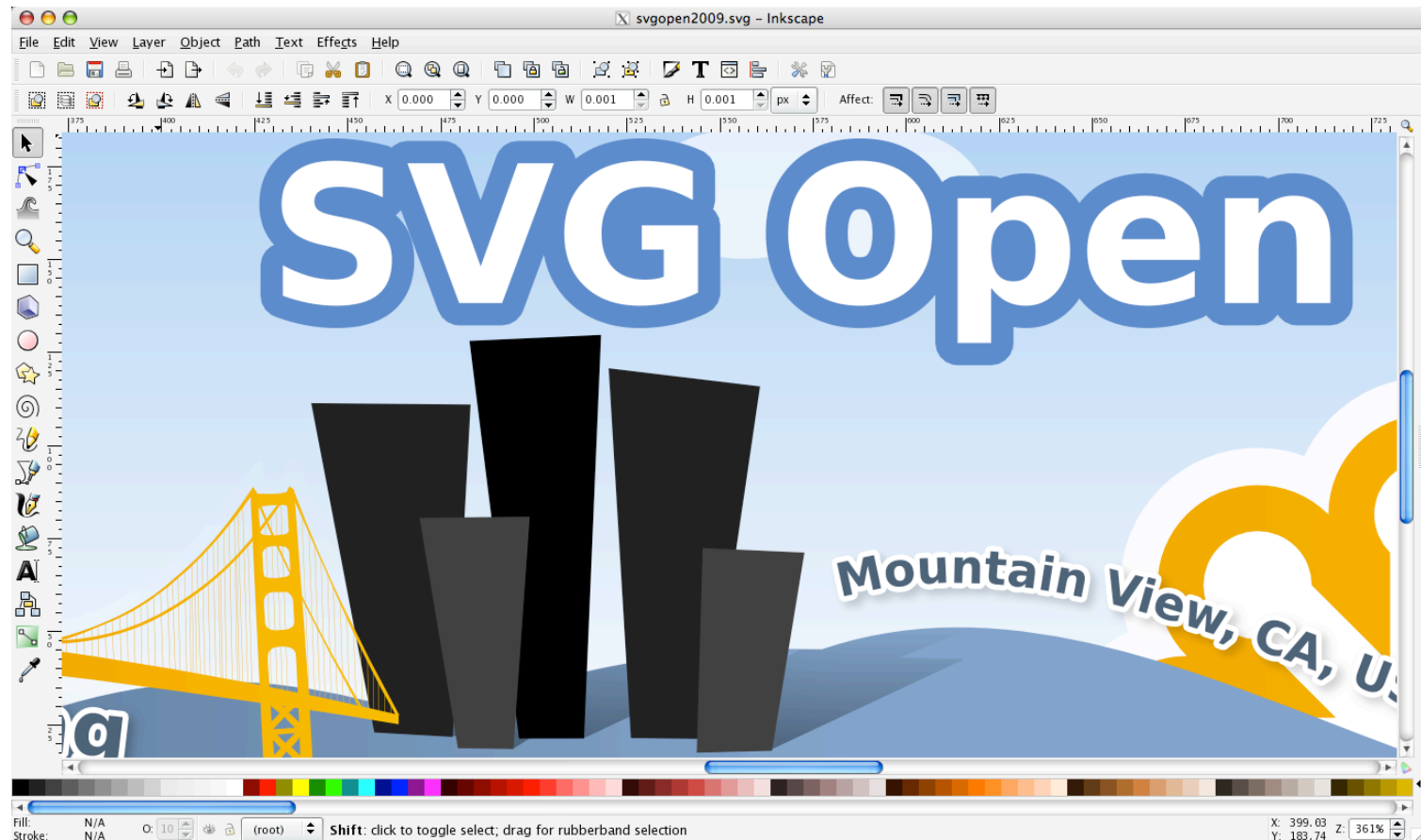
Browser Support (details)

Native Support	Firefox 1.5.0.11	2005-11-01		44.89%	F
	Firefox 2.0.1	2006-10-01		46.17%	F
	Firefox 3.0.0	2008-06-17		60.40%	C
	Firefox 3.5b+SMIL	2009-01-14		66.42%	C
	Firefox 3.5b Nightly	2009-03-29		60.40%	C
	Firefox pre3.6+SMIL	2009-03-29		67.52%	C
	Opera 8.5	2005-09-01		47.45%	F
	Opera 9.10	2006-12-01		89.96%	A
	Opera 9.50	2008-06-12		94.16%	A+
	Opera 10a1	2008-06-12		93.98%	A+
	Amaya 10	2008-02-26		27.45%	F
	Amaya 11	2008-12-16		28.55%	F
	Konqueror 3.5.5	2006-12-01		53.28%	D
	Konqueror 4.2.1	2009-03-04		29.64%	F
	Chrome 0.2	2008-09-02		61.50%	C
	Chrome 1.0	2008-12-01		61.86%	C
	Chrome 2.0 Nightly	2009-03-29		81.39%	A
	Safari 3 Beta	2007-06-01		52.74%	D
	Safari 3.1	2008-03-18		63.32%	C
	Safari 3.1.1	2008-04-16		62.96%	C
	Safari 3.2	2008-11-24		64.23%	C
	Safari 4 Beta	2009-02-24		81.93%	A
	WebKit r39960	2009-01-17		80.66%	A
	IE 7	2006-10-18		0.00%	F
	IE 8	2009-03-19		0.00%	F

Authoring



INKSCAPE

<http://www.inkscape.org/>

What's New?

- > IE Shim layer <http://code.google.com/p/sgweb/>
- > Conference



Demos

- > All demos at <http://delicious.com/chanezon/j1+svg>
- > More at <http://delicious.com/chanezon/svg+demos>

SVG Resources



INKSCAPE



- > <http://delicious.com/chanezon/svg>
- > <http://www.w3.org/Consortium/Offices/Presentations/SVG/0.svg>

When to use Canvas or SVG

Not competing

SVG

- retained mode graphics
- Editable static images
- Accessibility
- High-quality printing
- Interaction
- Mixing markupText

<canvas>

- immediate mode graphics
- Script-based scene graph
- Programmatic generation of images
- Drawing pixels
- Constant performance

Combining them possible in some browsers

HTML5 Resources

- > <http://blog.whatwg.org>
- > <http://code.google.com/doctype>
- > <https://developer.mozilla.org/En>
- > <http://dev.opera.com/>
- > <http://webkit.org/blog/>
- > Gmail mobile HTML5 blog posts
- > <http://delicious.com/chanezon/openweb>

Conclusion

"The Web has won -- it's the dominant programming model of our time"

Vic Gundotra, Google

- > HTML5 is already here
- > ... see <http://www.quirksmode.org/> for definition of "here"
- > If you want to know more about Canvas and SVG come to our talk at 6 pm: "2D vector graphics, the openweb way"



JavaOneSM

Thank You



Mark Pilgrim
Patrick Chanezon
chanezon@google.com
Google, Inc.

