



Java is a trademark of Sun Microsystems, Inc.

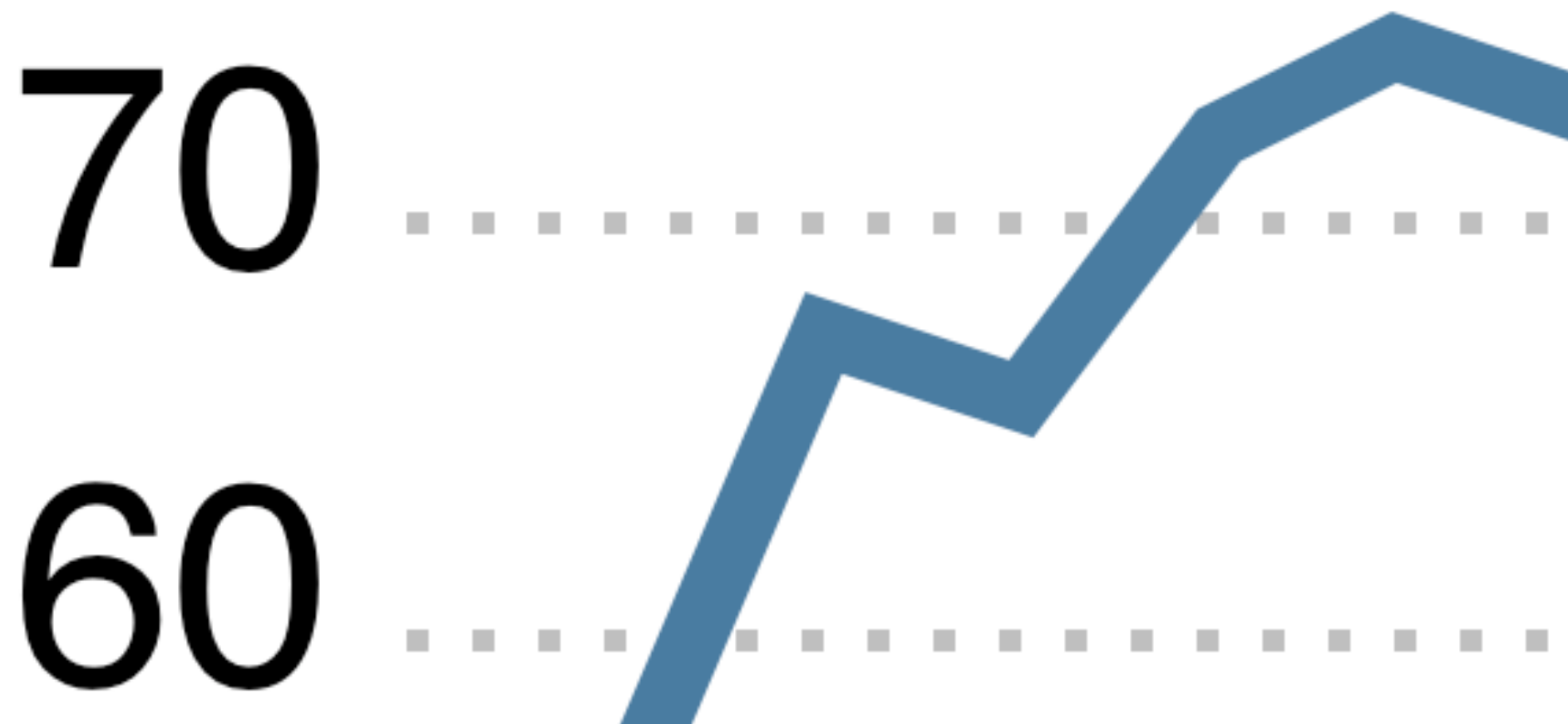


JavaOneSM

Cross-Browser Vector Graphics with the Canvas Tag and SVG

Patrick Chanezon
Ignacio Blanco
Google

Why Web Vector Graphics?



Why Web Vector Graphics?

- > Math specifications of shapes vs Array of pixels
 - Extremely small file sizes (SVG only): 50K JPG -> ~2K SVG file (Gzipped)
 - Resolution independence: from iPhone to 40" monitors
 - Use existing browser skills: JavaScript, DOM, CSS
 - Deep integration: no plugins, can interact with elements on the screen (video, layout, ...)
 - Looks great: smooth antialiasing
 - Full text markup (SVG only)
 - Accessible friendly
 - SEO Friendly
 - Archive Friendly

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%

W3C[®] WORLD WIDE WEB
c o n s o r t i u m

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), 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

<http://a.deveria.com/caniuse/>

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		10.0
Future (2010 or later)	9.0	4.0	4.*	2.0	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		10.0
Future (2010 or later)	9.0	4.0	4.*	2.0	10.*

<canvas> tag and Context object

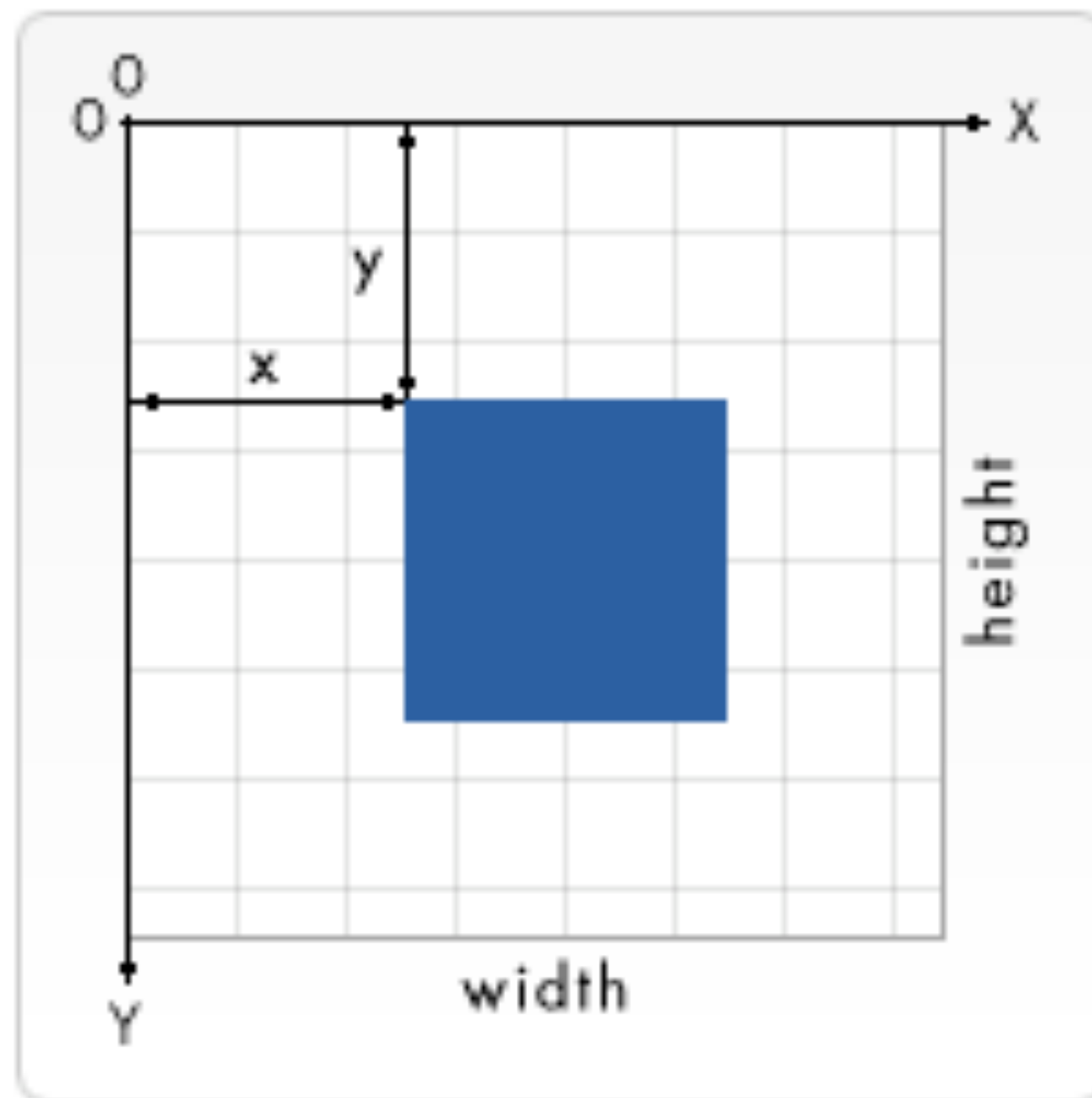
> Immediate mode graphics: fire and forget

```
<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);  
}
```

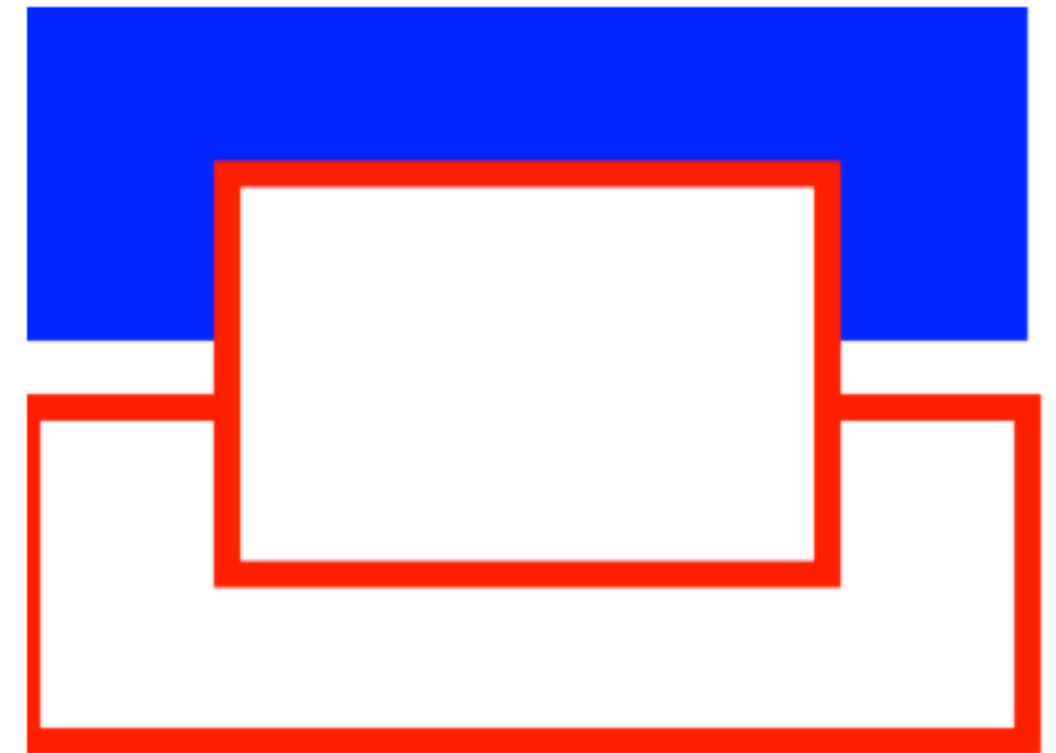
The Grid



fill, stroke, lines, Rect

```
context.fillStyle    = '#00f'; // blue
context.strokeStyle  = '#f00'; // red
context.lineWidth    = 4;

// Draw some rectangles.
context.fillRect(0, 0, 150, 50);
context.strokeRect(0, 60, 150, 50);
context.clearRect(30, 25, 90, 60);
context.strokeRect(30, 25, 90, 60);
```



Path

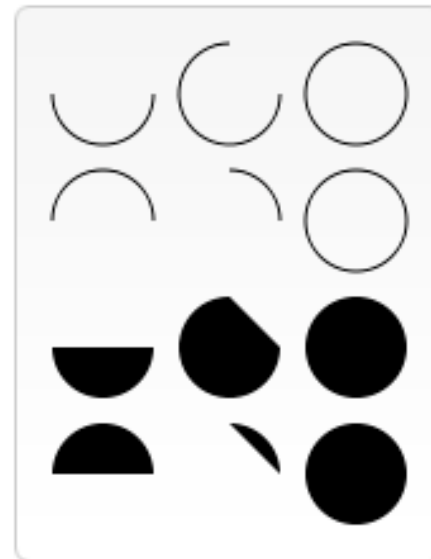
```
// Set the style properties.
context.fillStyle      = '#00f';
context.strokeStyle    = '#f00';
context.lineWidth      = 4;
context.beginPath();
// Start from the top-left point.
context.moveTo(10, 10); // give the (x,y) coordinates
context.lineTo(100, 10);
context.lineTo(10, 100);
context.lineTo(10, 10);

// Done! Now fill the shape, and draw the stroke.
context.fill();
context.stroke();
context.closePath();
```



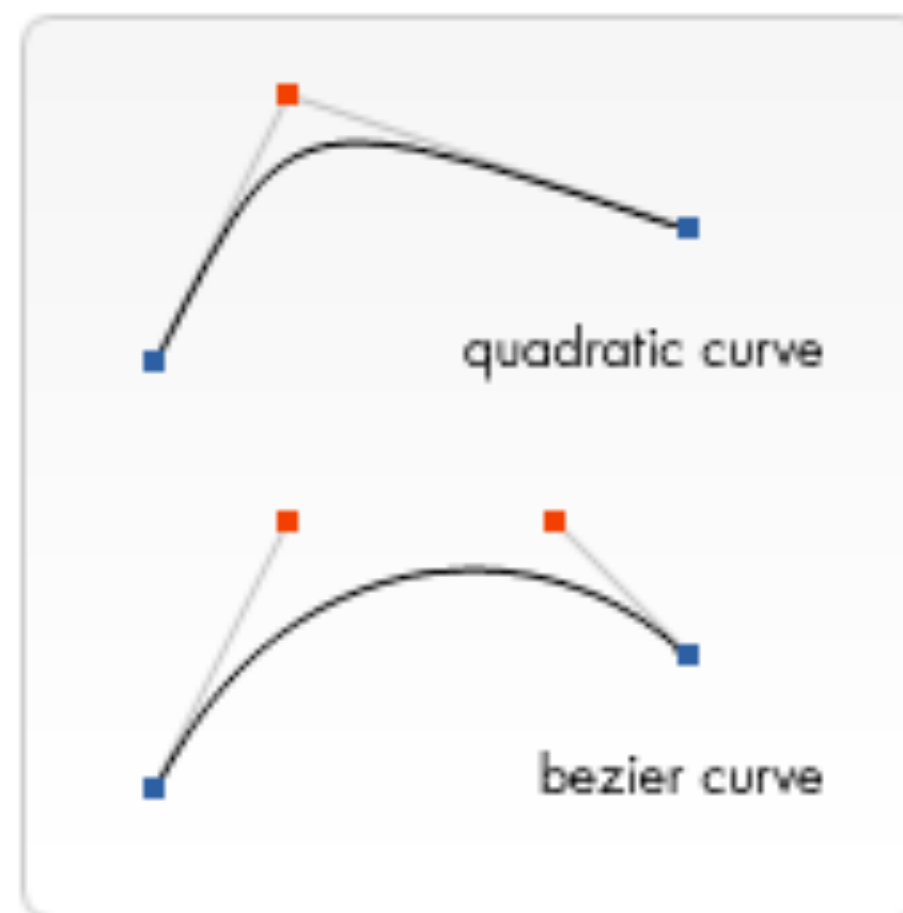
Arcs, Curves

`arc(x, y, radius, startAngle, endAngle, anticlockwise)`



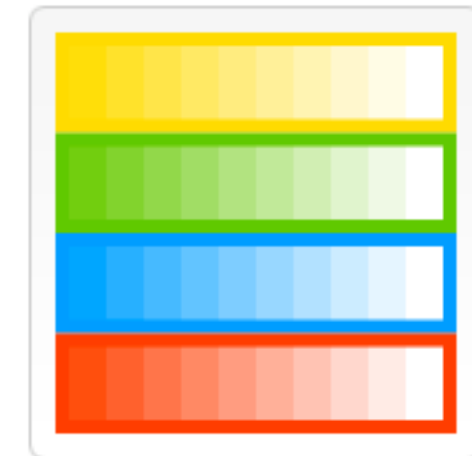
`quadraticCurveTo(cp1x, cp1y, x, y) // (BROKEN in FF 3.5)`

`bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)`

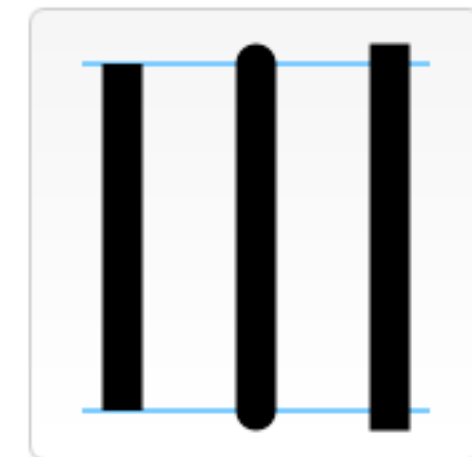


Styles and Colors

```
ctx.fillStyle = "orange";  
ctx.fillStyle = "#FFA500";  
ctx.fillStyle = "rgb(255,165,0)";  
ctx.strokeStyle = "rgba(255,0,0,0.5)";
```



```
lineWidth = value  
lineCap = type  
lineJoin = type  
miterLimit = value
```



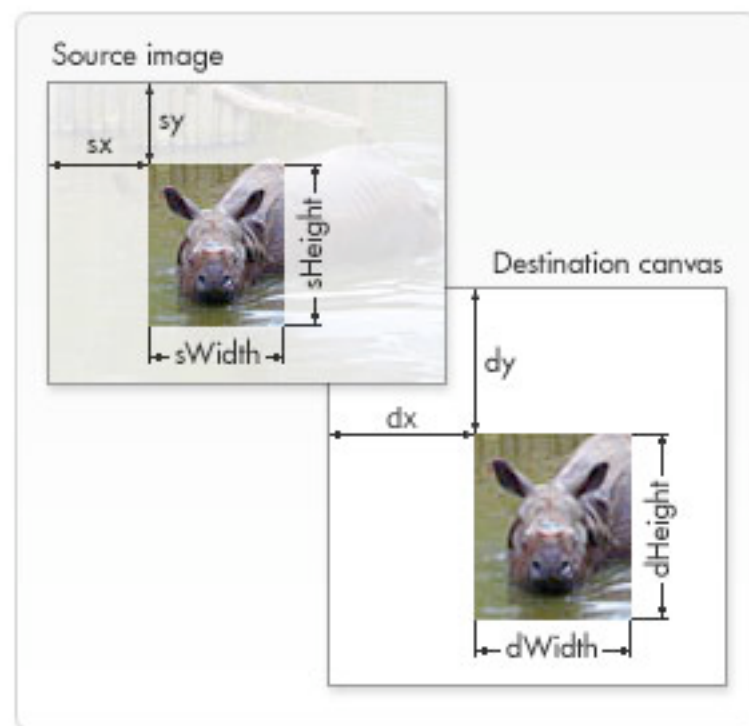
```
createLinearGradient(x1,y1,x2,y2)  
createRadialGradient(x1,y1,r1,x2,y2,r2)  
addColorStop(position, color)
```



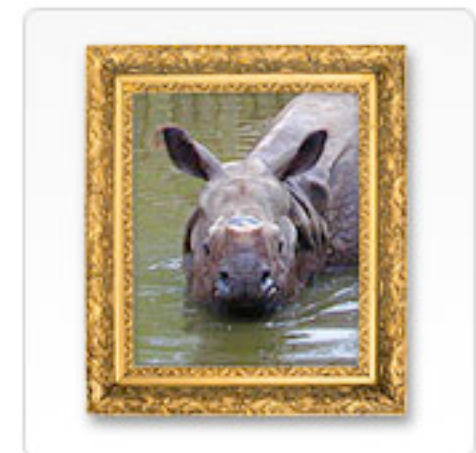
```
createPattern(image,type)
```


Images: draw, scale, slice

```
//drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)  
// Draw slice  
ctx.drawImage(document.getElementById( 'source' ),  
              33,71,104,124,21,20,87,104);
```

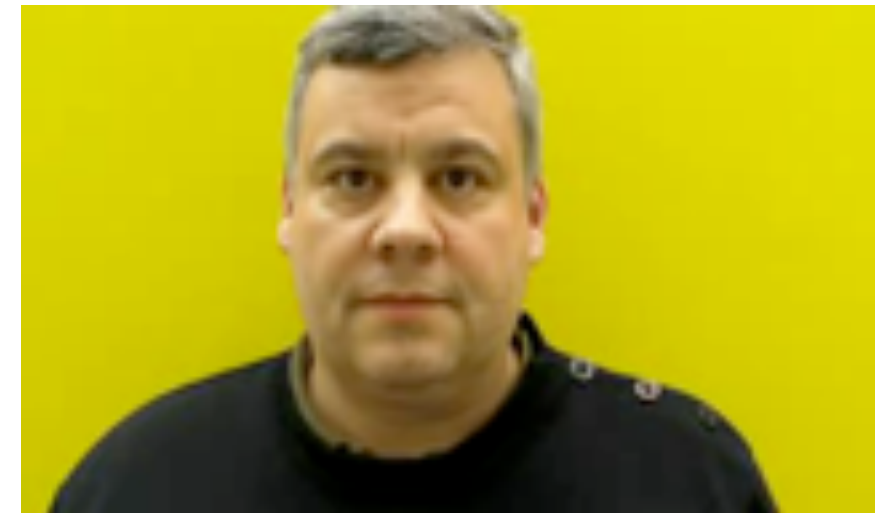


```
// Draw frame  
ctx.drawImage(document.getElementById( 'frame' ),0,0);  
context.stroke();  
context.closePath();
```



Pixel manipulation of Images, Video

```
computeFrame: function() {  
    this.ctx1.drawImage(this.video, 0, 0, this.width, this.height);  
    let frame = this.ctx1.getImageData(0, 0, this.width, this.height);  
    let l = frame.data.length / 4;  
  
    for (let i = 0; i < l; i++) {  
        let r = frame.data[i * 4 + 0];  
        let g = frame.data[i * 4 + 1];  
        let b = frame.data[i * 4 + 2];  
        if (g > 100 && r > 100 && b < 43)  
            frame.data[i * 4 + 3] = 0;  
    }  
    this.ctx2.putImageData(frame, 0, 0);  
    return;  
}
```



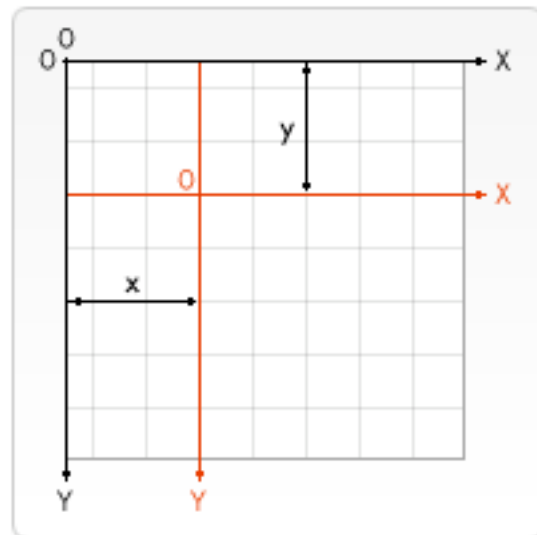
Transformations

State: Canvas states are stored on a stack

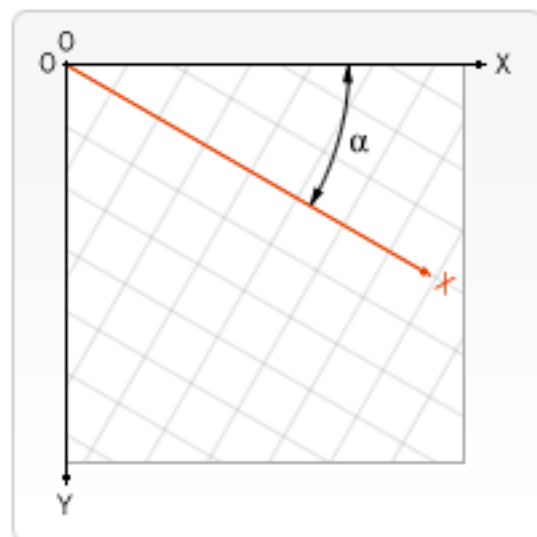
`save()`

`restore()`

`translate(x, y)`



`rotate(angle)`



`scale(x, y)`

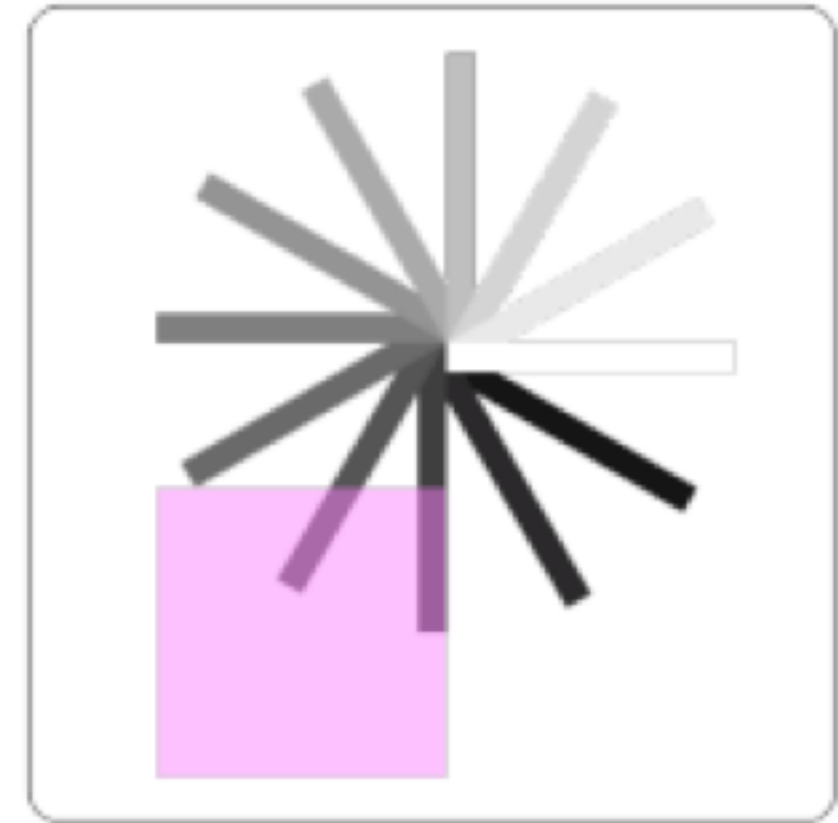
`transform(m11, m12, m21, m22, dx, dy)`

`setTransform(m11, m12, m21, m22, dx, dy)`

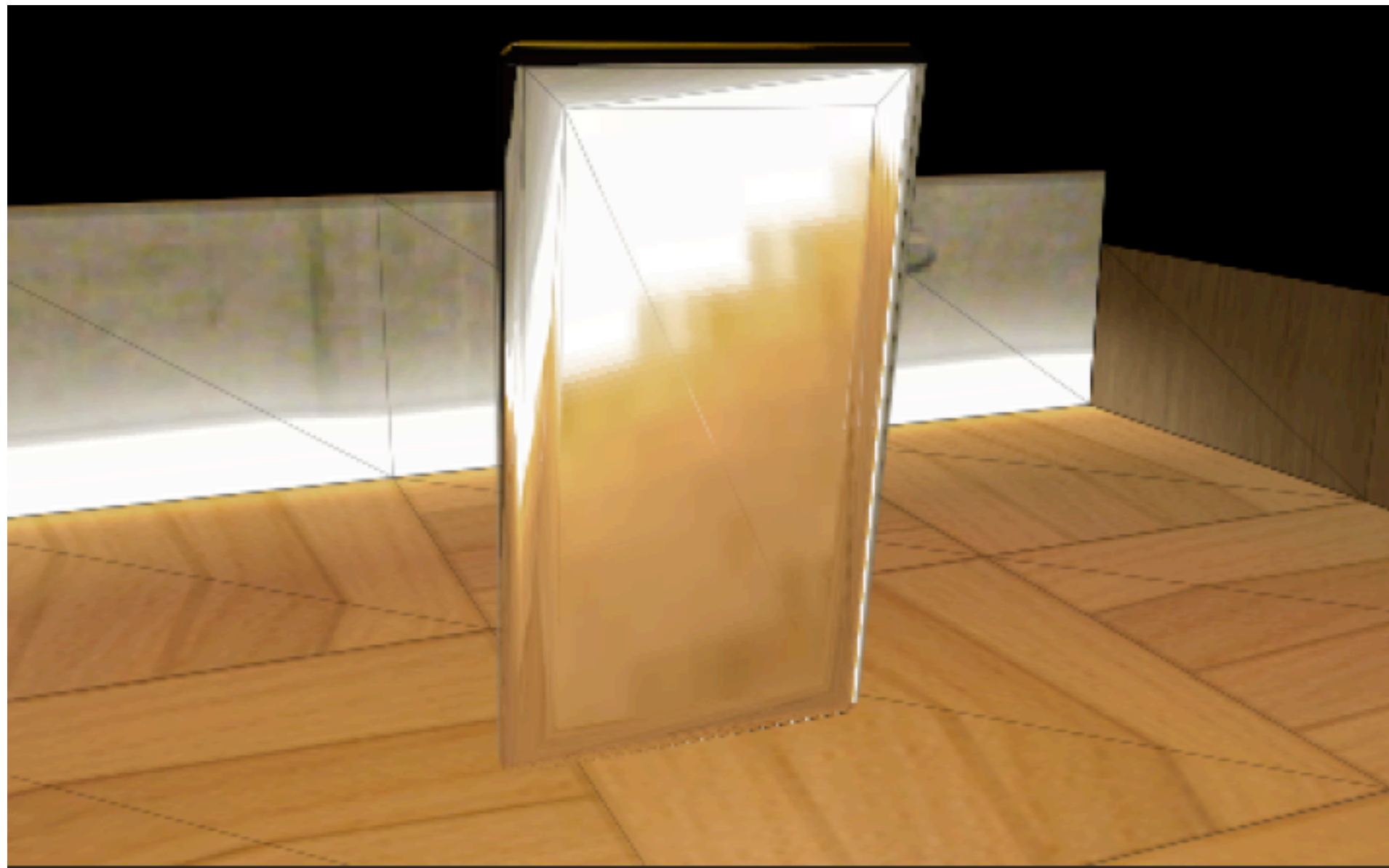
transform example

```
var i = 0;
var sin = Math.sin(Math.PI/6);
var cos = Math.cos(Math.PI/6);
ctx.translate(200, 200);
var c = 0;
for (i; i <= 12; i++) {
    c = Math.floor(255 / 12 * i);
    ctx.fillStyle = "rgb(" + c + "," + c + "," + c + ")";
    ctx.fillRect(0, 0, 100, 10);
    ctx.transform(cos, sin, -sin, cos, 0, 0);
}

ctx.setTransform(-1, 0, 0, 1, 200, 200);
ctx.fillStyle = "rgba(255, 128, 255, 0.5)";
ctx.fillRect(0, 50, 100, 100);
```



2D -> Triangles -> 3D



<http://kawanet.blogspot.com/2009/02/incredible-javascriptcanvas-3d-demos.html>

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](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: Text, Fonts, Animation, some Filters

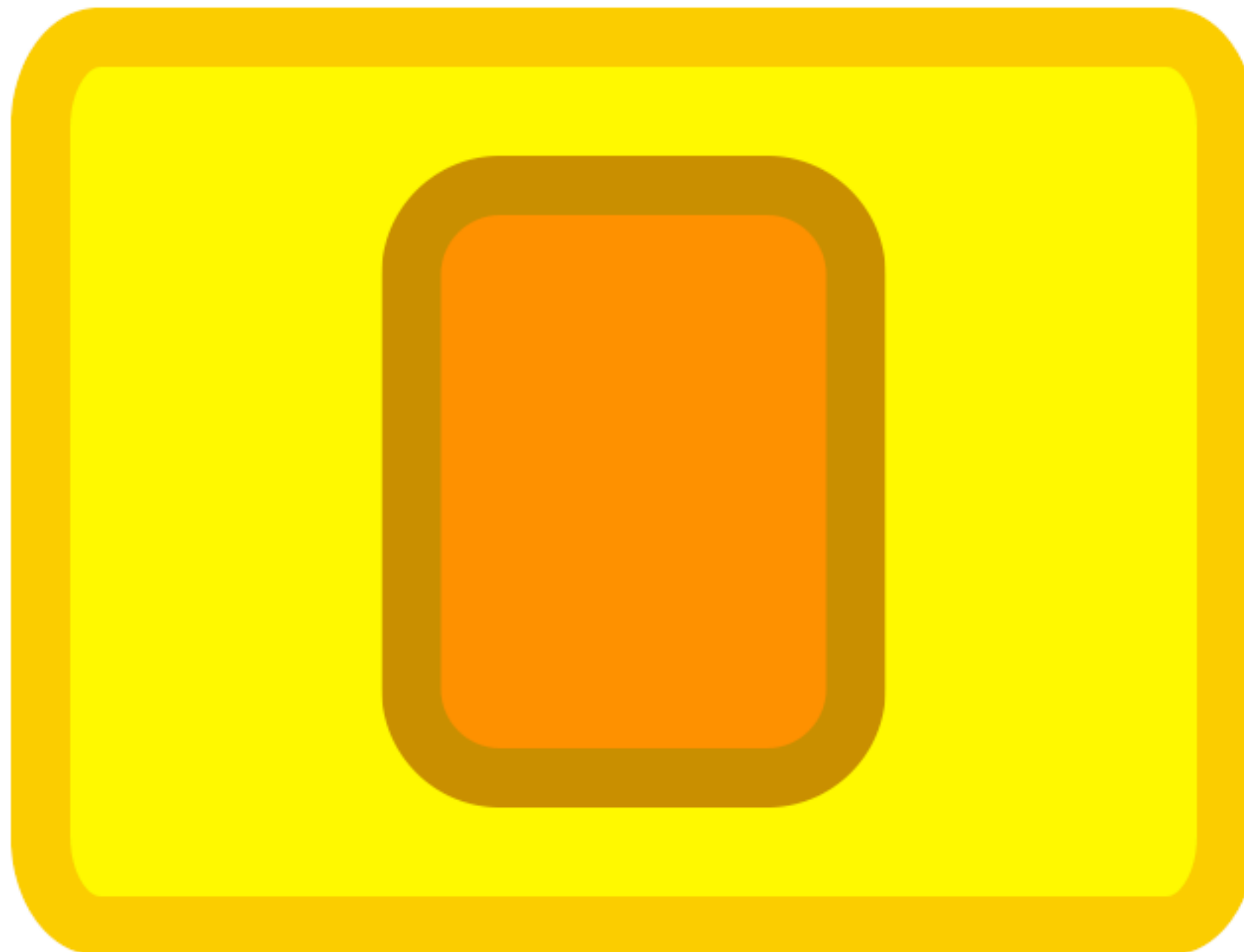
Coordinates, Viewbox

```
<?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>
```



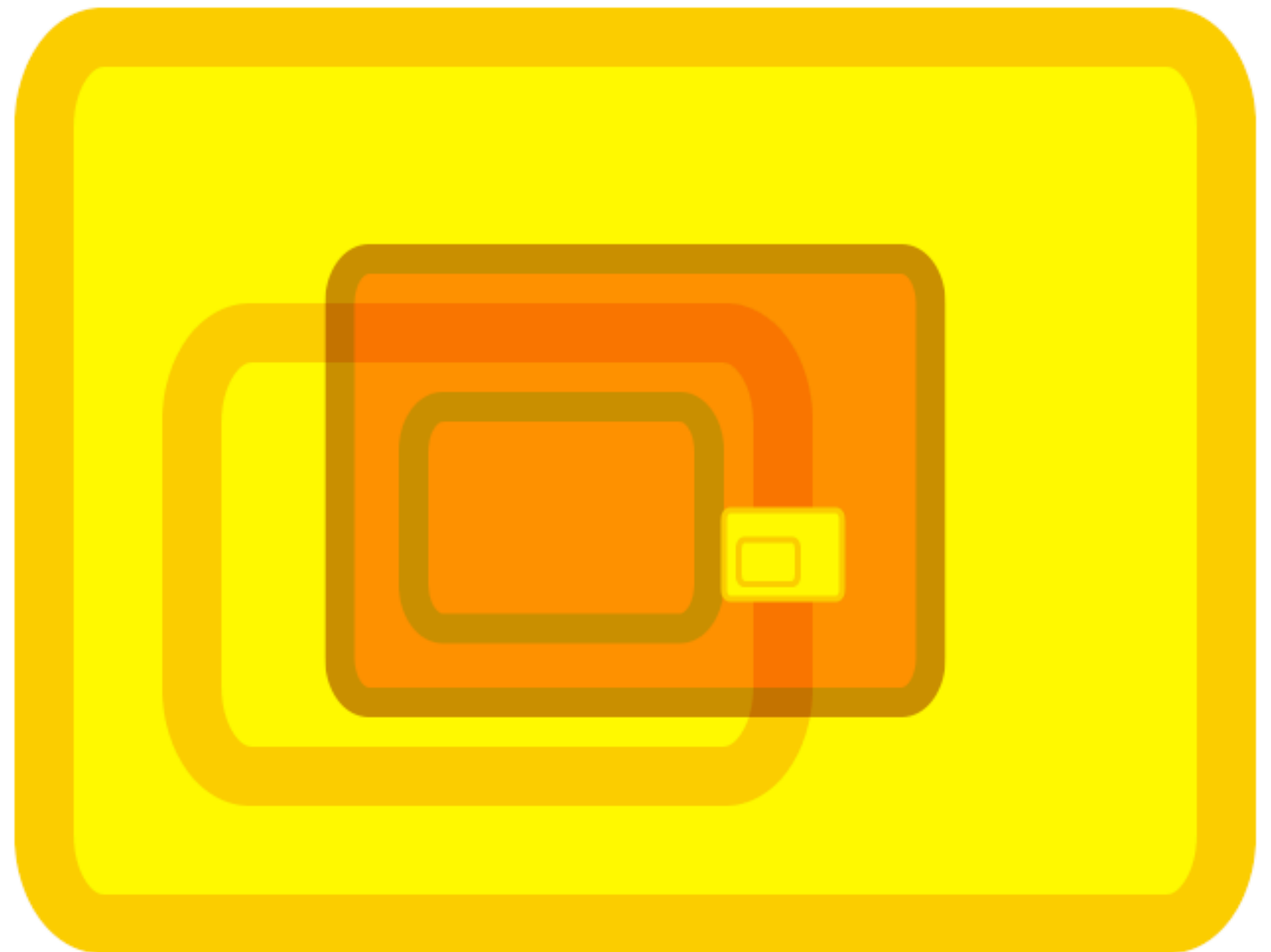
Transformations

```
<rect  
  width="200" height="150" rx="30" ry="30"  
  x="100" y="100" id="rect1"  
  style="opacity:0.5;fill:#ff0000;stroke:#800000;stroke-width:  
20;stroke-opacity:70"  
  transform="translate(100, 150) rotate(-90) translate(-300, 25)"/>
```



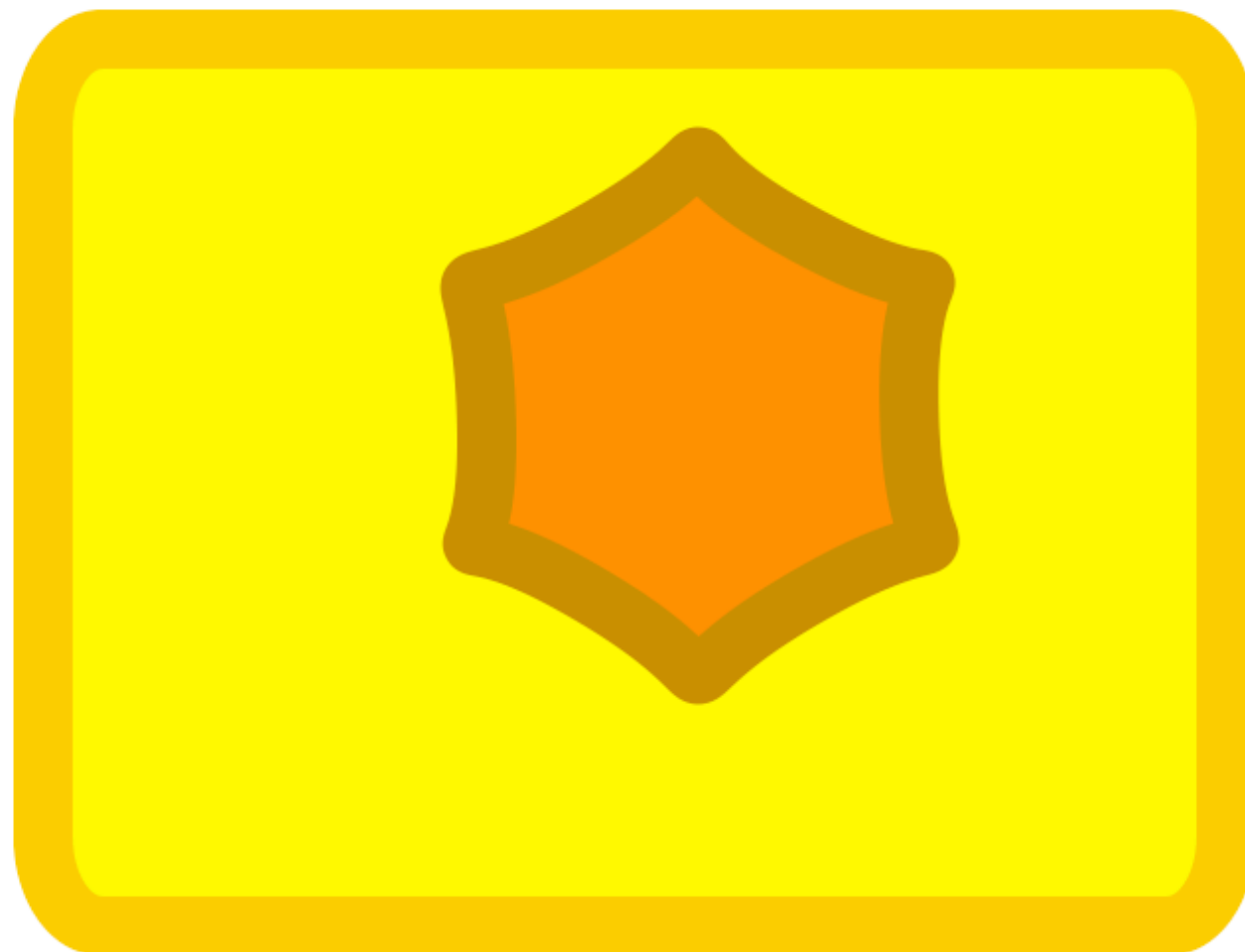
Grouping, Templates

```
<defs>
  <g id="layer1">
    <rect id="rect1"
      width="400" height="300"
      x="100" y="100"
      rx="20" ry="30" />
    <rect id="rect2"
      width="200" height="150"
      x="150" y="200"
      rx="20" ry="30" />
  </g>
</defs>
<use xlink:href="#layer1" style="fill:#fffc00;stroke:#fac305;stroke-width:
20" />
<use xlink:href="#layer1" transform="scale(0.5) translate(300, 250)"
style="opacity:0.5;fill:#ff0000;stroke:#800000;stroke-width:20;stroke-
opacity:70" />
<use xlink:href="#layer1" style="fill:#fffc00;stroke:#fac305;stroke-width:
20" transform="scale(0.1) translate(3200, 2500)" />
```



Path

```
<path
  d="M 336.68898,...237.69626 z"
  id="star"
  style="opacity:
0.5;fill:#ff0000;stroke:#800000;stroke-width:20;stroke-
opacity:70"
  transform="translate(20, 50) rotate(-10)"/>
```



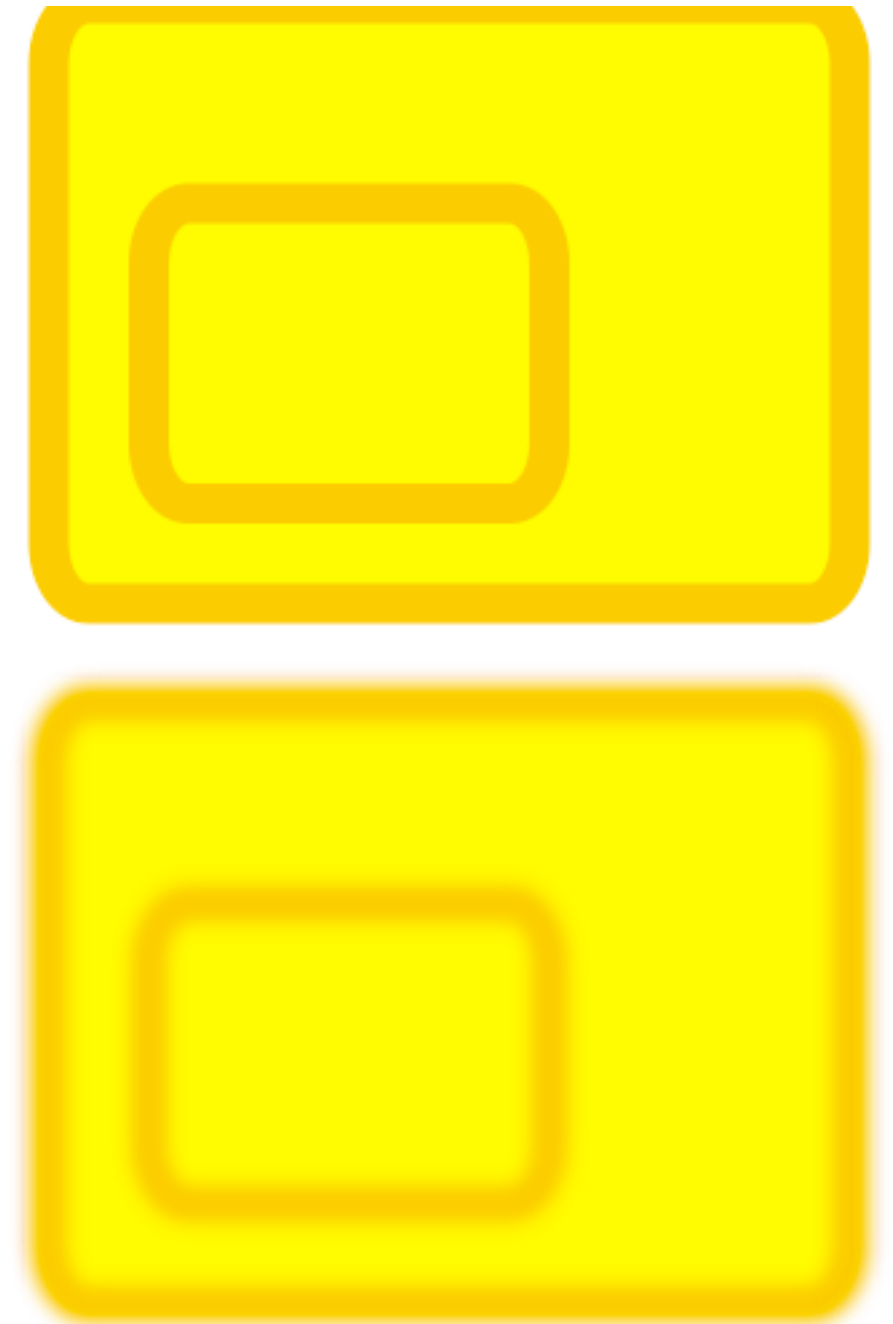
Text

```
<path
  d="M 198, ...,235.8809"
  id="spiral"
  style="opacity:0.0" />
<text style="font-size:30;font-family:Verdana;
fill:black">
  <textPath xlink:href="#spiral">Text spiraling out of
control!!</textPath>
</text>
```



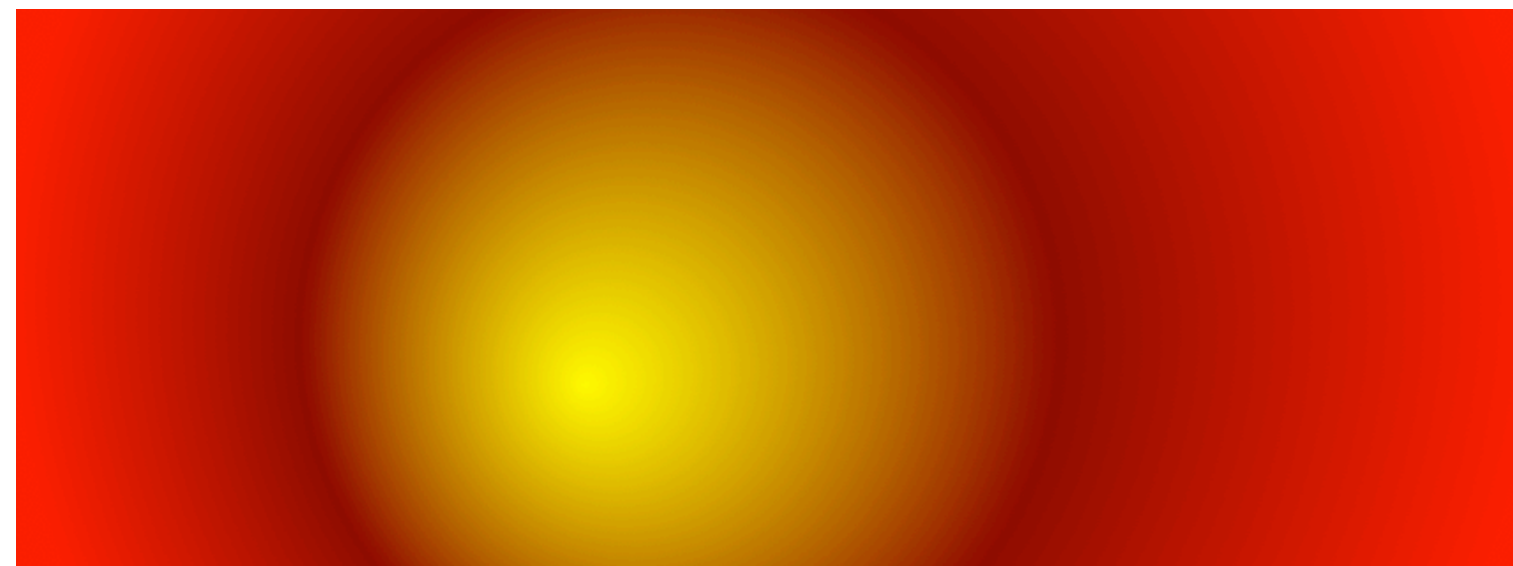
Filters

```
<defs>
  <filter id="drop-shadow">
    <feGaussianBlur
in="SourceGraphic" stdDeviation="4" />
  </filter>
  <g id="layer1" style="..."
...
  </g>
</defs>
<use xlink:href="#layer1" />
<use xlink:href="#layer1"
transform="translate(0, 350)"
filter="url(#drop-shadow)" />
```



Gradients

```
<defs>
  <radialGradient gradientUnits="userSpaceOnUse"
    id="MyGradient2" cx="200" cy="75"
    r="200" fx="150" fy="100">
    <stop offset="0" style="stop-color:#fffc00"/>
    <stop offset="0.5" style="stop-color:#800000"/>
    <stop offset="1" style="stop-color:red"/>
  </radialGradient>
</defs>
<rect x="0" y="0" width="400" height="150"
style="fill:url(#MyGradient2)"/>
```



Dom Scripting

```
<script type="text/ecmascript">
  <![CDATA[
    function showRectColor() {

alert(document.getElementById( "myBlueRect" ).getAttributeNS( null, "fill" ));
    }

    function showRectArea(evt) {
      var width = parseFloat(evt.target.getAttributeNS( null, "width" ));
      var height = parseFloat(evt.target.getAttributeNS( null, "height" ));
      alert("The rectangle area is: " + (width * height));
    }

    function showRootChildrenNr() {
      alert("Nr of Children: "+document.documentElement.childNodes.length);
    }
  ]]>
</script>
```

Other features

- > Clipping, masking
- > Patterns
- > CSS Styling
- > SVG Fonts
- > ...


























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

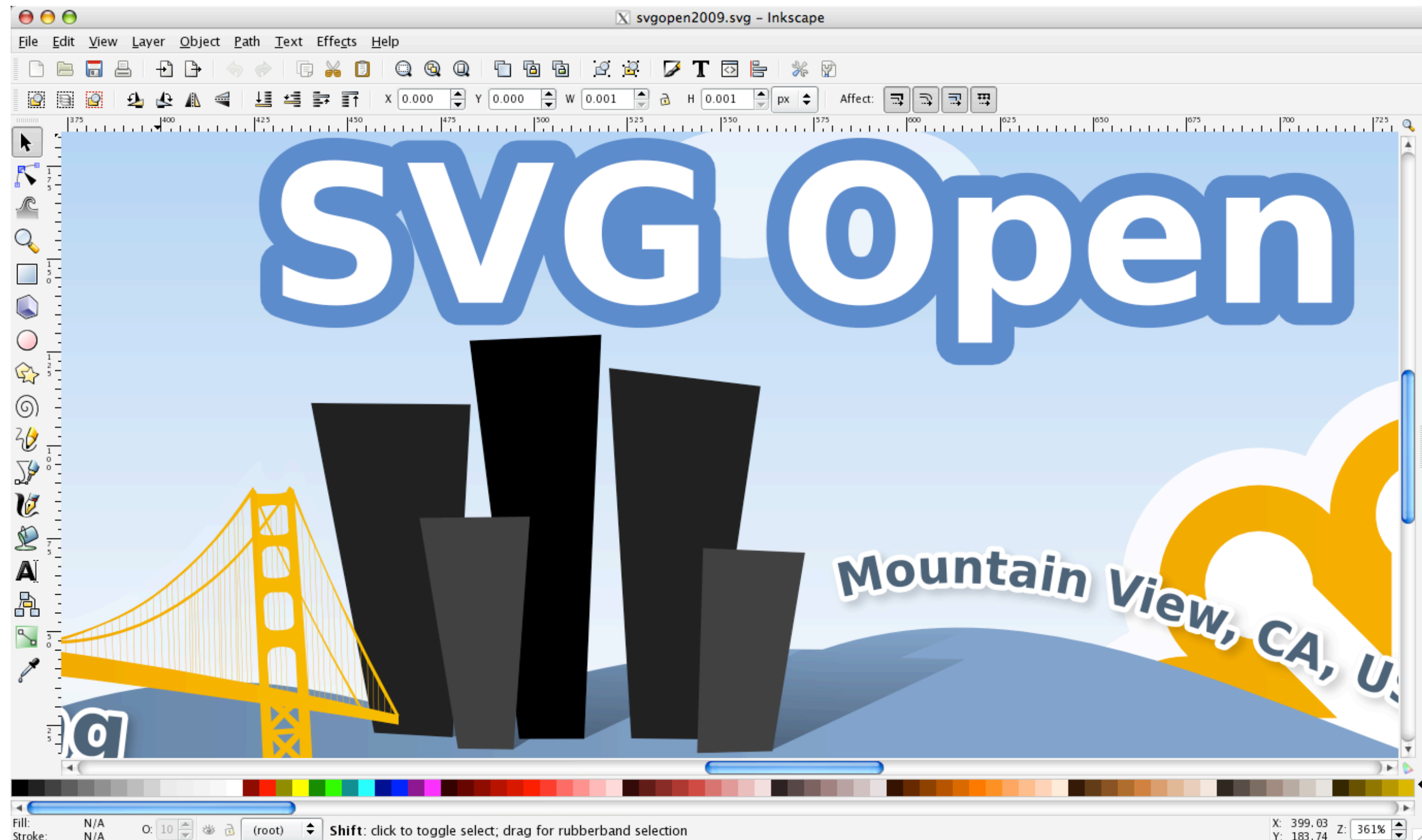
Browser Support (details)

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 markup
- Text

`<canvas>`

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

Combining them possible in some browsers

Example: using vector graphics from a Java Google Appengine application

- > Uses the following technologies
 - AppEngine Java
 - GWT
 - Google Chart API
 - GData API: Google Spreadsheets
 - Guardian Data API



JavaOneSM

Thank You

Patrick Chanezon
chanezon@google.com

Ignacio Blanco
blanconet@google.com

