

SVG in html5 and JavaScript

What are vector graphics?

There are two general ways to store an image in a digital system - pixel-based bitmaps and vector graphics.

In a pixel-based image (such as Windows bitmap, gif, jpeg, png, tiff files) the image is made of a set of dots (picture elements, pixels) in rows and columns. Each pixel contains red, green and blue components, and maybe a transparency value for layered images. Actual image file formats are like this (tiff is very simple), but maybe with techniques for compressing the file size (for pngs, gifs and jpegs).

In vector graphics, the image is made of a set of elements, such as lines, rectangles, circles and so on. Each element has attributes like size, position, fill colour, line colour, line thickness and so on. An vector graphics file simply lists these lines, circles and so on.

Vector graphics have a long history, back to the 1960s, used in applications like air traffic control and CAD.

SVG

SVG is 'scalable vector graphics'.

This usually means W3C's specification of an XML-based vector graphics mark-up language. The current version is 1.1. It has shapes named path, rect, circle, ellipse, line, polyline and polygon.

When to use SVG

Pixel-based storage is better for photos, paintings and similar. To describe a photo in terms of simple circles, rectangles and lines would need a very large number of very small circles and so on, to get an accurate image. SVG for photos takes an enormous amount of storage.

But for line drawings, diagrams, charts and similar images SVG is much better. A chart is typically made of a few rectangles and lines, and this is just a small number of elements, so the SVG memory size is small.

Also, SVG is *scalable* - this is a key feature.

Suppose you have a jpeg image with resolution 10 by 10 pixels, and you want to enlarge it to 100 X 100 pixels. You need some way to fill in the missing pixels (interpolation) and this never works well. Or you can just enlarge the pixels you have, producing a blocky 'pixellated' image.

But for a SVG image you just scale up the sizes and positions of the lines and rectangles and other elements. It looks just as good when you zoom in or out.

A third advantage of SVG in a webpage is that you can access, and modify, the picture programmatically - usually with JavaScript. So you can create and alter the image by program code inside a browser.

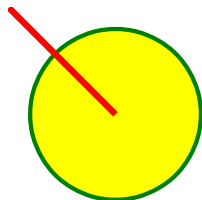
SVG in html5

html5 has two new graphics elements - `<canvas>` and `<svg>`. Canvas is used to display 2d and 3d pixel graphics. svg display SVG graphics.

So a web page can have an `<svg>` element (or several). Inside the svg we can have a set of image elements, such as

```
<!DOCTYPE html>
<html>
  <head>
    <title>SVG</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body >
    <svg width="100" height="100"> <!-- SVG element -->
      <!-- a circle -->
      <circle cx="50" cy="50" r="40" stroke="green" stroke-width="2" fill="yellow" />
      <!-- a line -->
      <line x1="0" y1="0" x2="50" y2="50" stroke="red" stroke-width="3"/>
    </svg> <!-- end SVG -->
  </body>
</html>
```

Which renders as



SVG in JavaScript

The SVG element is in the DOM, so JavaScript lets us manipulate the DOM using standard techniques.

We can create suitable shapes, set its attributes, then append it to the DOM in the appropriate place.

So for a basic page:

```
<!DOCTYPE html>
<html>
  <head>
    <title>SVG</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script>
      function draw()
      ..
    </script>
  </head>
  <body>
  </body>
</html>
```

```
</script>
</head>
<body onload="draw()">
  <svg id="mySVG" width="100" height="100">
    </svg>
</html>
```

The JavaScript could be

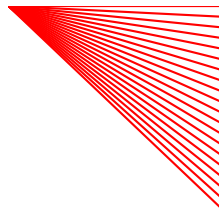
```
function draw()
{
  var svgPart = document.getElementById("mySVG");
  // remove anything already there
  while (svgPart.firstChild) {
    svgPart.removeChild(svgPart.firstChild);
  }

  var svgns = "http://www.w3.org/2000/svg"; // XML namespace

  for (var y = 0; y < 100; y += 5)
  {
    var shape = document.createElementNS(svgns, "line");
    shape.setAttributeNS(null, "x1", 0);
    shape.setAttributeNS(null, "y1", 0);
    shape.setAttributeNS(null, "x1", 100);
    shape.setAttributeNS(null, "y1", y);

    shape.setAttributeNS(null, "stroke", "red");
    svgPart.appendChild(shape);
  }
}
```

which draws:



Links

The w3c SVG 1.1 document: <http://www.w3.org/TR/SVG/expanded-toc.html>

MDN's SVG pages: <https://developer.mozilla.org/en-US/docs/Web/SVG>

From Jenkov: <http://tutorials.jenkov.com/svg/scripting.html>