Scalar Vector Graphics

Table of Contents

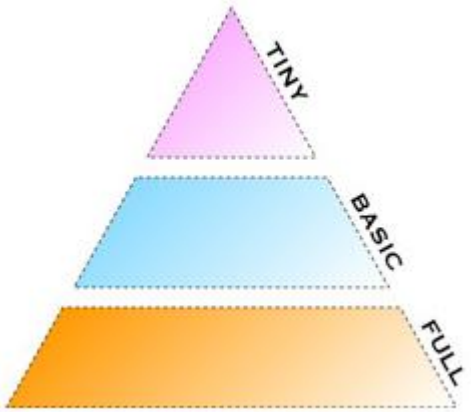
1. Introduction

SVG is used to describe vector images . It is used for constructing images using XML. SVG can be manipulated through code results in re-rendering.

SVG can be delivered as a standalone file, but its initial focus is integration with HTML. It is a part of HTML document and has its own elements, attributes and styles.

1.1 History

SVG (retained mode model) development began in 1999 as a response from big Companies such as Microsoft, Apple and IBM had their hands in defining and creating this new feature. In 2001, SVG got a W3C recommendation to include mobile profiles, print capabilities and improved display properties. There were constant talks about what after SVG 1.0 which concentrated on old desktop-type machine, this lead to the emergence of SVG Mobile. SVG PDA and SVG Mobile needed something smaller to fit into it. In 2001 many leading companies contributed to produce SVG Tiny and SVG Basic called SVG Mobile. As the mobile phones have progressed a great deal and consumers and service providers were looking for greater and richer graphic experience. W3C working group came to use the new work done in SVG 1.2. SVG 1.2 Tiny doesn't just adds features from SVG 1.1 but has its own new features. While it supports <audio> element, talks are on for support for <video> element.



1.2 Working of SVG

SVG works withing XML . It requires :-

1.2.1 Declaration Statement

Consists of XML declaration statement, version, encoding and standalone.

<?xml version="1.0" encoding="UTF-8" standalone= "yes"?>

1.2.2 Root Element

Parent element of all the elements.

Example:

<root>

    <child>

    </child>

<root>

1.2.3 Namespaces

An abstract web address called Uniform Resource Identifier distinguishes namespaces.

Example:

<?xml version="1.0" ?>

<colortheory>

<orange xmlns="http://www.mycolorpages.com/colortheory">

<approach> Mix red and yellow pigments together </approach>

</orange>

</colortheory>

1.3 SVG Color Attributes

SVG uses one of the three color codes to draw shapes with XML.

- **Rgb Value** - stands for red, green and blue to specify a shade. Rgb(0,0,255) is the key value for blue.
- **Color Name** - giving color name: blue
- **Hexadecimal Value** - assigned numerical code that defines each color: #0000FF is blue

2. SVG Tools

2.1 Illustrator, Inkscape

These are vector editor tools. You can simply create your own images on these tools and save file as SVG. You can see SVG code apart from extra meta data and add it into HTML for rendering it on browser.

2.2 Raphael.js

Raphael js is a lightweight javascript library that creates SVG on web. As SVG does not support IE 8 and below, it will convert the code to VML.

Example:

```
<html>
<head>
<script type="text/javascript" src="C:/Markup Project/Raphael/raphael.js"></script>
<script type="text/javascript" src="C:/Markup Project/Raphael/script.js"></script>
                    <script src='https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.min.js'></script>
<script>
    $(function () {
      var paper = Raphael(10, 50, 300, 250);
      var circle = paper.circle(50, 40, 10);
      circle.attr('fill', '#c00');
      circle.attr('stroke', '#fff');
    });
</script>
</head>
                    <body>
          <title>Raphael Play</title>
<div id="canvas_container"></div>
```

```
</body>

</html>
```

Resulting Image:

3. SVG Filters

SVG filters are used to give images a nice effect.

3.1 SVG Filters Input and Output

SVG filters can take Graphic, Alpha Channel or output of another Filter as input.

Example:

```
<defs>
<filter id="blurFilter2" y="-10" height="40" x="-10" width="150">
<feOffset in="SourceAlpha" dx="3" dy="3" result="offset2" />
<feGaussianBlur in="offset2" stdDeviation="3"  result="blur2"/>

<feMerge>
<feMergeNode in="blur2" />
<feMergeNode in="SourceGraphic" />
</feMerge>
</filter>
</defs>

<ellipse cx="55" cy="60" rx="25" ry="15"
      style="stroke: none; fill: #0000ff; filter: url(#blurFilter2);" />
```

Resulting Image:

3.2 Blend Filter

Blend Filter blends input from multiple filters into one.

Example:

```
<svg width="500" height="100">
<defs>
<filter id="blurFilter3" y="-10" height="40" x="-10" width="150">
<feOffset in="SourceAlpha" dx="3" dy="3" result="offset3" />
<feGaussianBlur in="offset3" stdDeviation="3"  result="blur3"/>

<feBlend  in="SourceGraphic" in2="blur3" x="-10" width="160"/>

</filter>
</defs>

<ellipse cx="55" cy="60" rx="25" ry="15"
        style="stroke: none; fill: #0000ff;
        filter: url(#blurFilter3);" />

</svg>
```

Resulting Image:

4. Gradients

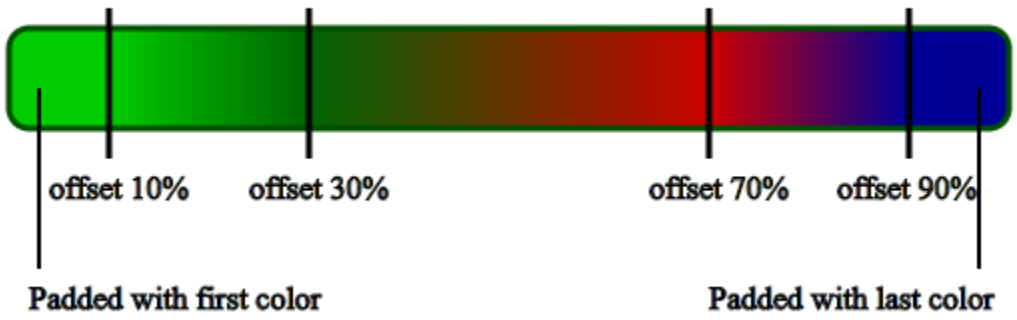
SVG gradients fill shapes with color in an uneven fashion. Fill or stroke of a shape changes from one color to another.

SVG gradients are of two types:

4.1 Linear Gradient

Linear gradients changes color in vertical, horizontal or along a vector in an evenly linear fashion.

```
<svg xmlns="http://www.w3.org/2000/svg"
     xmlns:xlink="http://www.w3.org/1999/xlink">
<defs>
<linearGradient id="myLinearGradient1"
          x1="0%" y1="0%"
          x2="100%" y2="0%"
          spreadMethod="pad">
<stop offset="10%" stop-color="#00cc00" stop-opacity="1"/>
<stop offset="30%" stop-color="#006600" stop-opacity="1"/>

<stop offset="70%" stop-color="#cc0000" stop-opacity="1"/>
<stop offset="90%" stop-color="#000099" stop-opacity="1"/>

</linearGradient>

</defs>

<rect x="10" y="10" width="500" height="50" rx="10" ry="10"
        style="fill:url(#myLinearGradient1); stroke: #005000;
        stroke-width: 3;" />

</svg>
```



4.2 Radial Gradient

Gradients change in a circular fashion in Radial Gradient.

Example:

```
<svg xmlns="http://www.w3.org/2000/svg"
     xmlns:xlink="http://www.w3.org/1999/xlink">

<defs>
```

```
<radialGradient id="myRadialGradient4"
        fx="5%" fy="5%" r="65%"
        spreadMethod="pad">
<stop offset="0%"   stop-color="#00ee00" stop-opacity="1"/>
<stop offset="100%" stop-color="#006600" stop-opacity="1" />
</radialGradient>
</defs>

<rect x="340" y="10" width="100" height="100" rx="10" ry="10"
      style="fill:url(#myRadialGradient4);
      stroke: #005000; stroke-width: 3;" />
</svg>
```

Resulting Image:

5. SVG Shapes

SVG has some predefined shapes.

5.1 SVG Rectangle

Example:

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">
<rect x="4" y="10" width="200" height="200" style="fill: green; stroke:#006600">
</rect>
</svg>
```

Resulting Image:

5.2 SVG Circle

Example:

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">
<circle id="circle1" r="20" cx="40" cy="100" fill="tomato">
</circle>
</svg>
```

Resulting Image:

5.3 SVG Ellipse

Example:

```
<svg height="140" width="500">

<ellipse cx="200" cy="80" rx="80" ry="40" style="fill:tomato;stroke:purple;stroke-width:2" />

</svg>
```

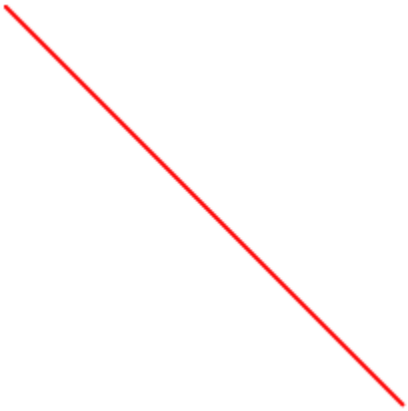
Resulting Image:

5.4 SVG Line

Example:

```
<svgheight="210"width="500">
<linex1="0"y1="0"x2="200"y2="200"style="stroke:rgb(255,0,0);stroke-width:2"/>
</svg>
```

Resulting Image:

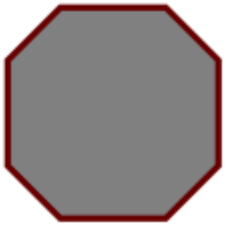


5.5 SVG Polygon

Example:

```
<svg xmlns="http://www.w3.org/2000/svg"
   xmlns:xlink="http://www.w3.org/1999/xlink">

<polygon points="50,5   100,5  125,30  125,80 100,105
      50,105  25,80  25, 30"
      style="stroke:#660000; fill: gray; stroke-width: 3;"/>

</svg>
```

Resulting Image:



5.6 SVG Polyline

Example:

```
<svg xmlns="http://www.w3.org/2000/svg"
   xmlns:xlink="http://www.w3.org/1999/xlink">

<polyline points="10,2  60,2  35,52"
      style="stroke:#006600; stroke-width: 2;
      fill: tomato;"/>
</svg>
```

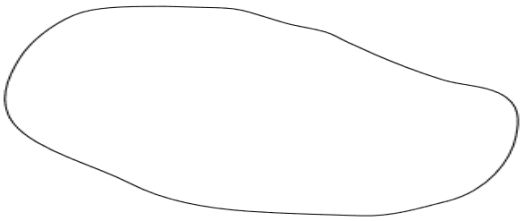
Resulting Image:

5.7 SVG Path

Example:

```
<path id="motionPath" fill="none" stroke="#000000" stroke-miterlimit="10" d="M202.4,58.3c-13.8,0.1-33.3,0.4-44.8,9.2
      c-14,10.7-26.2,29.2-31.9,45.6c-7.8,22.2-13.5,48-3.5,70.2c12.8,28.2,47.1,43.6,68.8,63.6c19.6,18.1,43.4,26.1,69.5,29.4
      c21.7,2.7,43.6,3.3,65.4,4.7c19.4,1.3,33.9-7.7,51.2-15.3c24.4-10.7,38.2-44,40.9-68.9c1.8-16.7,3.4-34.9-10.3-46.5
      c-9.5-8-22.6-8.1-33.2-14.1c-13.7-7.7-27.4-17.2-39.7-26.8c-5.4-4.2-10.4-8.8-15.8-12.9c-4.5-3.5-8.1-8.3-13.2-11
      c-6.2-3.3-14.3-5.4-20.9-8.2c-5-2.1-9.5-5.2-14.3-7.6c-6.5-3.3-12.1-7.4-19.3-8.9c-6-1.2-12.4-1.3-18.6-1.5
      C222.5,59,212.5,57.8,202.4,58.3"/>
```

Resulting Image:



5.8 SVG Text

Example:

```
<svg xmlns="http://www.w3.org/2000/svg"
   xmlns:xlink="http://www.w3.org/1999/xlink">

<text x="20"  y="20">Example SVG text</text>

</svg>
```

Resulting Image:

Example SVG text

6. Benefits

- **SVG Text Based -** SVG images are created and edited using text editor. While using graphic editor, an image is required to produce an image . SVG works differently. Software program is required to draw a picture and the final code is collection of words.
- **Resolution Independent** - Resolution-based issues is a hot topic for the front end developers. Most of the time unnecessary data download is required for a better resolution. SVG offers a high quality resolution, no matter what size, what zoom level or resolution. Images in SVG looks same at any resolution.
- **No unnecessary HTTP Requests** - With the use of <img> tag in HTML, you are defining an image which user browser has to request and that would take up lot of bandwidth. But instead if you use DOM-node based API of SVG would cut extra HTTP request and makes website faster and user friendly. It is good for mobile devices as it uses less number of HTTP request.
- **Bandwidth friendly -** No matter how large the graphic sets get, as it is described inside a XML, graphic is the only thing transmitted to the client. From millimeters to meters bandwidth requirement remains the same.
- **SVG 1.2 Tiny enhancements -** Text wrapping and non-scaling strokes are available for rectangular regions. Recent phones have the capability of <audio> element for playback of sound.
- **Other Benefits -** Text labels, descriptions that can be searched, interactive custom events and animation with the help of scripting.

7. Drawbacks

- **Complex Development** - Although SVG requires any text editor for writing code, but at times simple shape can lead to many lines of code which becomes difficult to troubleshoot for errors.

Learning SVG

- **Lack of Information** - You can find a lot of information on HTML and XML but with SVG there is not much material available.
- **Limited Authoring Tools** - Though many tools are available for SVG, but developers find it difficult and time consuming on coding SVG on text editors. More software tools are available for other format which saves time.
- **Browser Support** - Once you have created SVG image, it requires substantial testing on various browsers. Enhanced SVG animation have poor level of browser support. Thus making it less suitable for animated games. Loading of heavy SVG files takes time.
- **Converting SVG -** Converting an image to SVG will never be to a pixel level. Client devices crashes if the image is too complicated.

8. Practical Applications

- **Graph** - SVG is used for creating static graphs from vector coordinates and live graphs using AJAX requests. Google maps consists of SVG.
- **Road Map** - SVG's strength is vector shapes. Vector graphics represent hard lines and exact shapes which does not distort on zooming.
- **Complex UI Elements** - Instead of using CSS and HTML, you could use predefined shapes in SVG to create a complex UI Element or with the help of Illustrator Tool one can save it out as an SVG file.
- **Logos -** Most of the logos are vector based.
- **Simple Games** - SVG can be greatly used in the field of gaming as it requires less character animation and more information display. Example Sudoku.

9. References

- http://webdesign.about.com/od/svg/a/what-is-svg.htm
- http://msdn.microsoft.com/en-us/library/ie/ff972259(v=vs.85).aspx
- http://msdn.microsoft.com/en-us/library/aa288462(v=vs.71).aspx
- https://inkscape.org/en/learn/faq/
- http://raphaeljs.com/reference.html
- https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute
- http://msdn.microsoft.com/en-us/library/ie/gg193983%28v=vs.85%29.aspx
- http://ledrug.wordpress.com/2010/09/30/learning-svg/
- http://tutorials.jenkov.com/svg/switch-element.html
- http://code.tutsplus.com/tutorials/an-introduction-to-the-raphael-js-library--net-7186
- https://developer.mozilla.org/en-US/docs/Web/SVG/Element/tspan

Learning SVG

- http://css-tricks.com/guide-svg-animations-smil/
- http://www.hongkiat.com/blog/scalable-vector-graphics-animation/
- http://oak.is/thinking/animated-svgs
- http://code.tutsplus.com/articles/why-arent-you-using-svg--net-25414
- http://www.informit.com/articles/article.aspx?p=99036&seqNum=2
- https://seanmacentee.com/how-and-why-you-should-use-svg-images-on-your-site/
- http://creative-jar.com/insights/digital-technology/why-use-svg/
- http://www.xml.com/pub/a/2004/06/16/mobilesvg.html
- http://www.w3.org/Graphics/SVG/About.html
- http://alistapart.com/article/using-svg-for-flexible-scalable-and-fun-backgrounds-part-i
- http://www.appnovation.com/blog/benefits-scalable-vector-graphics-svg