

Basic SVGs

Eduardo Navas (ean13@psu.edu)

The basic code for the following instructions can be found in the book [*Interactive Data Visualization*](#) by Scott Murray

Notes on editing SVGs.

SVG stands for Scalable Vector Graphic. This means that the image consists of mathematical proportions that can be made smaller or bigger without loss of detail or quality.

SVGs are useful to develop apps, games, data visualizations, and miscellaneous images for the web that are simple in color.

SVGs are optimal for D3 projects. This type of image is used to develop all types of visualizations from simple barcharts to complex mappings of countries around the world.

To create an SVG is simple. Here is code for a square. This code is embedded between the tags `<body></body>` of an html page:

```
<svg width = "1000" height = "1000">  
  
<rect x="0" y="0" width="100" height="100" fill="purple"/>  
  
</svg>
```

Notice that we have a tag for svg: `<svg></svg>`. You can think of this tag as the canvas where you will create your image composition. You can make the svg area as small or large as you like. Just be aware that some of your information or design may appear beyond the pre-defined range if you make the objects bigger or place them beyond the overall size of the SVG canvas.

Next, notice that we have the attributes for a rectangle. This begins with “rect” followed by coordinate of x and y, which are allocated the value of “0.” Next you can adjust the height and width by putting a number in quotes. In the example above, this number is “100” pixels for both coordinates. And finally, you can give the rectangle a color by using the attribute “fill,” which in this case is purple.

You can change the position, height, and color by inputting another number to the attributes described above. The html example that comes along with this tutorial gives you the following group of squares:

```
<svg width = "1000" height = "1000">
```

```
<rect x="0" y="0" width="100" height="100" fill="purple"/>
<rect x="20" y="5" width="100" height="100" fill="blue"/>
<rect x="40" y="10" width="100" height="100" fill="green"/>
<rect x="60" y="15" width="100" height="100" fill="yellow"/>
<rect x="80" y="20" width="100" height="100" fill="red"/>

</svg>
```

Notice how the squares move down-right by adjusting the x and y. You can make similar adjustments to the other attributes to develop a composition of your own.

We can also do circles. Here is one circle:

```
<svg width = "1000" height = "1000">

<circle cx="60" cy="280" r="60" fill="rgba(128, 0,128, 1.0)"/>

</svg>
```

Notice that the x y axis are now written as cx and cy. Also, we have a radius, that defines the size of the circle. The fill can also be listed as a number as opposed to a name. In this case the rgba attribute is used to set red, green, and blue values along with the alpha for transparency. The numbers for color are from 255 to 0 and the alpha from 1.0 to .01. The colors and alpha are listed in sequence as shown in the example above.

The color attributes above are decimal code. You can find a full range at:
http://en.wikipedia.org/wiki/Web_colors

A composition of circles slightly adjusted in position and opacity would be coded like this:

```
<svg width = "1000" height = "1000">

<circle cx="60" cy="280" r="60" fill="rgba(128, 0,128, 1.0)"/>
<circle cx="100" cy="320" r="60" fill="rgba(0, 0,255, 0.75)"/>
<circle cx="140" cy="360" r="60" fill="rgba(0,255, 0, 0.5)"/>
<circle cx="180" cy="400" r="60" fill="rgba(255, 255, 0, 0.25)"/>
<circle cx="220" cy="440" r="60" fill="rgba(255, 0, 0, 0.1)"/>

</svg>
```

You can also create ellipses as follows:

```
<ellipse cx="200" cy="80" rx="100" ry="50" style="fill:yellow;stroke:purple;stroke-width:2" />
```

```
<ellipse cx="200" cy="200" rx="100" ry="50" style="fill:#FF00FF;stroke:purple;stroke-width:10" />
```

Notice the difference in this case is that there is now an rx and an ry attribute to give the ellipses their oval shapes. Also note that the fill color is assigned in different ways from previous examples. All of these ways of writing the attributes are fine. If you want to use a # color or call up a specific color such as “yellow” and you want to set the opacity to a different number other than 100% (1.0), as the ellipses above show, you will need to use the attribute, opacity. It would look like this:

```
opacity = “0.3”
```

This would go inside the corresponding tag of rect, circle, ellipse, along with other attributes.

Make sure to view the example available with this tutorial for SVGs (“Basic_SVGs.html”). The page will also include variations of the placements of circles and squares.