SVG Basics

In the last couple D3 classes we've learned a bit about SVGs and the SVG coordinate system. This quiz is meant to review some of those core concepts. To complete this quiz, open up the "SVGQuiz.html" doc in your text editor and in a web browser. In this quiz, we won't be using D3 at all -- we'll simply be drawing shapes in the browser. Below the quiz questions, you'll find more in-depth information on SVGs (this might be helpful to you as you complete the quiz).

Implementing SVGs in HTML: To create an SVG and add shapes using basic HTML, first you add an SVG element and define its width and height. The following is the code for an SVG that has width 100 pixels wide and height 100 pixels. In this example the background color is gray, but often you'll work with a transparent SVG element and add colored shapes.

```
<svg width="100" height="100" style="background-color:gray"></svg>`
```

To add shapes (circles, lines, polygons, etc.) you add them within the <svg> tags. For example, you can add a circle as follows:

```
<svg width="600" height="150">
        <circle cx="200" cy="50" r="15"/>
</svg>
```

TASK 1: Add an SVG element that is 800 by 800 pixels.

TASK 2: Create a *very* simple bar chart by drawing five rectangles. Scale the rectangles vertically according to the following data points to illustrate state population (your bar should should be vertical, not horizontal) and use a fill color of #3CAEA3.

California - 39.5 million
Texas - 28.9 million
Florida - 21.4 million
New York - 19.4 million
Pennsylvania - 12.8 million

TASK 3: Add lines for a horizontal and vertical axis. Make the color gray and the stroke-width 2.

TASK 4: Add text labels for each state (abbreviations are fine).

SVG Review

About SVG: SVG stands for “Scalable Vector Graphic” and you can use SVGs to create graphics (shapes, lines, etc.) on websites. D3 visualizations are based on SVGs. For example, in D3 bar charts are based on a series of rectangles, scatter plots include a series of circles, line charts consist of a series of lines, etc. When you make a visualization in D3, you’ll be combining SVG shapes to make a compelling visual.

SVG Coordinate system: SVGs are based on a grid with X and Y coordinates. You assign X and Y values to determine the position of the shape, measured in pixels. Unlike the mathematical grid we’re used to, with SVGs the starting point (0, 0) is at the top left. When you create SVGs on a “canvas”, you always set the starting point for the shape based on coordinates. If you want a circle at the top left of the canvas, you set X = 0 and Y = 0. If you want a circle inset by 100 pixels both vertically and horizontally, you set X = 100 and Y = 100.

Adding Shapes

You can add a range of shapes to your SVG element, including rectangles, circles, ellipses, polygons and lines. You can also add text. To add shapes, first you set the X and Y coordinates where the shape starts. Remember, the coordinates start from the top left of your SVG element. If you set X = "10" and Y = "5" as the starting point of the circle, then the center of the circle will be positioned 10 pixels to the right and 5 pixels down from the top left of your SVG element.

After defining the X and Y position for the starting point of your shape, you define the characteristics of the shape itself. These characteristics depend on the shape. For a circle, you simply define the radius (r). For a rectangle, you define the width and height.

Here's an example of a rectangle positioned at (200,50) with a height of 50 pixels and a width of 75 pixels. Notice that since we don't define a background color for the SVG, it's transparent.

```
<svg width="600" height="150">
        <rect x="200" y="50" width="75" height="50" />
</svg>
```

You can change the color to yellow by adding the following code:

```
<svg width="600" height="150">
        <rect x="200" y="50" width="75" height="50" style="fill:yellow" />
</svg>
```


Adding Additional Shapes

You can add a variety of other shapes by adding them as html elements to your SVG. Here are some of them:

Rectangle (rect)

Circle (circle)

Line (line)

Text (text)

Ellipse (ellipse)

Polygon (polygon)

For example, here's the code for a circle with radius 15. Notice that the X and Y position is denoted with "cx" and "cy".

```
<svg width="600" height="150">
        <circle cx="200" cy="50" r="15"/>
</svg>
```

Adding Lines

Here's an example of code for a line that is 200 pixels tall and has a stroke-width of 5. To create lines, you define starting and end points for the line. The starting point is "x1" and "y1" and the ending point is "x2" and "y2".

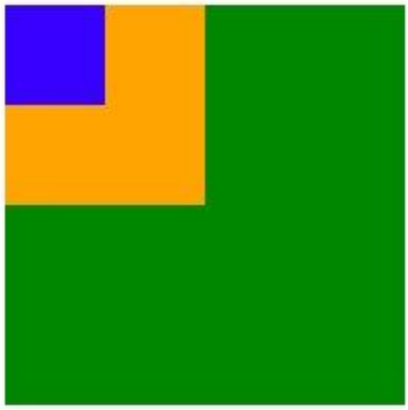
```
<svg width="600" height="400">
        <line x1="100" y1="100" x2="100" y2="300" style="stroke:black;stroke-width:5">
</svg>
```

Adding Styling

To change the color of shapes such as circles, add a "fill". To change outline colors, add a "stroke" and "stroke-width".

Here's an example of code for three squares positioned within one another. The outermost square has a gray stroke.

```
<svg width="600" height="400">
        <rect x="200" y="100" width="200" height="200" style="fill:green" />
        <rect x="200" y="100" width="100" height="100" style="fill:orange" />
        <rect x="200" y="100" width="50" height="50" style="fill:blue" />
</svg>
```



Ordering of SVGs

In the above example of three squares, take note of the layering of the squares. Now we'll replicate the same code, but change the order of the rectangles, putting the blue first, then orange, then green. View the results below.

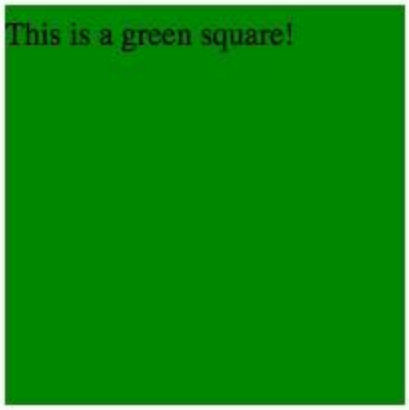
```
<svg width="600" height="400">
        <rect x="200" y="100" width="50" height="50" style="fill:blue" />
        <rect x="200" y="100" width="100" height="100" style="fill:orange" />
        <rect x="200" y="100" width="200" height="200" style="fill:green" />
</svg>
```

Notice how the green square appears but the other squares don't appear. This is due to the layering order of the rects. The 2nd and 3rd rects appear on top of the first rect, so the final green rect covers the first two. When you're putting shapes on a canvas, keep in mind the order you place them to make sure all shapes are visible to users.

Adding Text

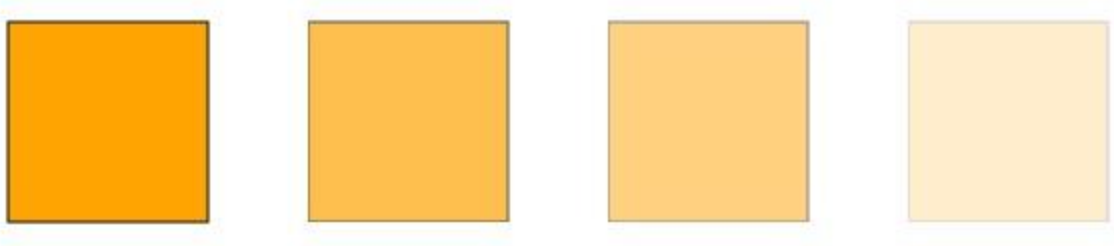
With SVGs, you can also add text. This can be useful for labeling items in a visualization. To add text, use the syntax in the code below:

```
<svg width="600" height="400">
        <rect x="200" y="100" width="200" height="200" style="fill:green" />
        <text x="200" y="120">This is a green square!</text>
</svg>
```



You can style SVGs in many different ways, such as adding strokes and fill colors and changing opacity. Below is an example of 5 rectangles styles with strokes and varying levels of opacity.

```
<svg width="700" height="400">
        <rect x="100" y="150" width="100" height="100"  style="stroke:black; opacity:1;
        fill:orange" />
        <rect x="250" y="150" width="100" height="100" style="stroke:black; opacity:.7;
        fill:orange" />
        <rect x="400" y="150" width="100" height="100" style="stroke:black; opacity:.5;
        fill:orange" />
        <rect x="550" y="150" width="100" height="100" style="stroke:black; opacity:.2;
        fill:orange" />
</svg>
```



To add labels to the boxes, add text elements to the SVG, as demonstrated below.

```
<svg width="700" height="400">
        <rect x="100" y="150" width="100" height="100"  style="stroke:black; opacity:1;
        fill:orange" />
```

```
        <rect x="250" y="150" width="100" height="100" style="stroke:black; opacity:.7;
        fill:orange" />
        <rect x="400" y="150" width="100" height="100" style="stroke:black; opacity:.5;
        fill:orange" />
        <rect x="550" y="150" width="100" height="100" style="stroke:black; opacity:.2;
        fill:orange" />
        <text x="100" y="270">Opacity: 1</text>
        <text x="250" y="270">Opacity: .7</text>
        <text x="400" y="270">Opacity: .5</text>
        <text x="550" y="270">Opacity: .2</text>
</svg>
```