## Drawing an SVG Straight Line using D3.js

We previously saw how to construct an SVG Straight Line in the [Basic Building Blocks](https://www.dashingd3js.com/basic-building-blocks) section:

Straight Line Example & Code:

1<svg width="50" height="50">

2 <line x1="5" y1="5" x2="40" y2="40" stroke="gray" stroke-width="5" />

3</svg>

Using the Circle, Rectangle, and Ellipse examples, you can probably guess that to build a straight line, you will need a "x1", "y1", "x2" and "y2".

You can then make the process of drawing a straight line as two step process:

1) Make an SVG container (the SVG Coordinate Space)

2) Draw the straight line

So in the simplest case, our JavaScript code would look like:

1*//Make an SVG Container*

2**var** svgContainer = d3.select("body").append("svg")

3 .attr("width", 200)

4 .attr("height", 200);

5

6*//Draw the line*

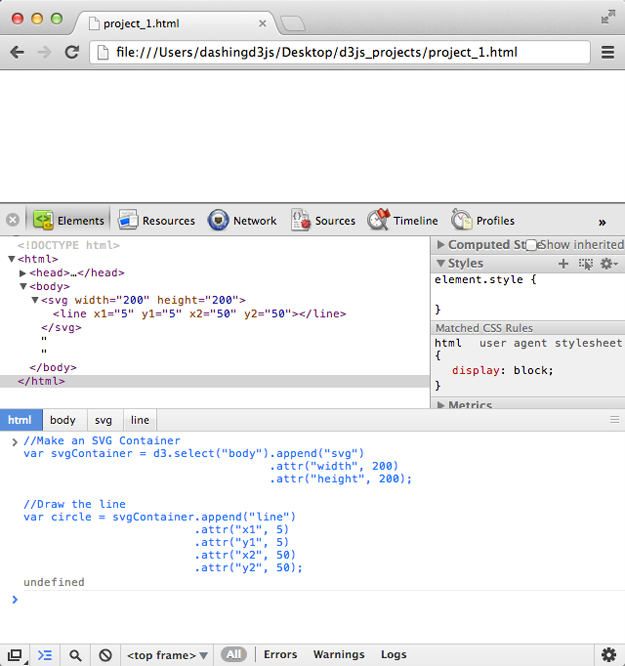
7**var** circle = svgContainer.append("line")

8 .attr("x1", 5)

9 .attr("y1", 5)

**10** .attr("x2", 50)

11 .attr("y2", 50);

Which gives us:   
  


**Wait a second?! Where is the line?????** The SVG element is there, however we can't see it in our browser.

It turns out because the SVG ‘line’ elements are single lines and thus are geometrically one-dimensional, that is - they have no interior.

Which means that ‘line’ elements are never filled (see the ‘fill’ property).

Which means that our line does not take up space - so we don't actually see anything.

To fix this, make sure to give the line:

* **.attribute("stroke-width", NUMBER)**, where NUMBER is how wide the line is in units
* **.attribute("stroke", "COLOR")**, where COLOR is a color to used to color the line

So fixing our simplest case, our JavaScript code would look like:

1 *//Make an SVG Container*

2**var** svgContainer = d3.select("body").append("svg")

3 .attr("width", 200)

4 .attr("height", 200);

5

6*//Draw the line*

7**var** circle = svgContainer.append("line")

8 .attr("x1", 5)

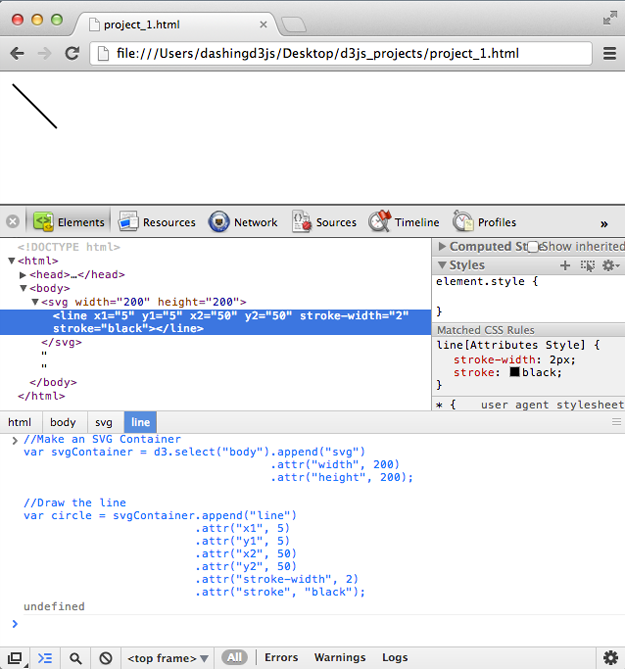
9 .attr("y1", 5)

**10** .attr("x2", 50)

11 .attr("y2", 50)

12 .attr("stroke-width", 2)

13 .attr("stroke", "black");

Which gives us:   
  


Fantastic - now the line is visible!

The necessary SVG attributes for drawing a straight line are the "x1", "y1", "x2", "y2", "stroke" and "stroke-width".

Note - We don't use a style method with the line. Because ‘line’ elements are single lines and thus are geometrically one-dimensional, they have no interior. Which is why to style them we need to deal with the "stroke" color and "stroke-width".

Notice that the important attributes we need to draw an SVG Straight Line in D3.js are - **x1**, **y1**, **x2**, **y2**,**stroke-width** and **stroke**.

## Drawing Polyline & Polygon SVG Basic Shapes using D3.js

The basic shapes we still have to cover are "Polyline" and "Polygon".

We previously saw how to construct an SVG Polyline and Polygon in the [Basic Building Blocks](https://www.dashingd3js.com/basic-building-blocks) section:

Polyline Example & Code:

1<svg width="50" height="50">

2 <polyline fill="none" stroke="blue" stroke-width="2"

3 points="05,30

4 15,30

5 15,20

6 25,20

7 25,10

8 35,10" />

9</svg>

Polygon Example & Code:

1<svg width="50" height="50">

2 <polygon fill="yellow" stroke="blue" stroke-width="2"

3 points="05,30

4 15,10

5 25,30" />

6</svg>

Using the Circle, Rectangle, Ellipse, and Straight Line examples, you can probably guess that to build a Polyline and Polygon, you will need a "stroke", "stroke-width" and "points". And "fill" for the Polygon.

However, as you can see from the examples above - the **points** attribute contains a list of points where the x and y are separated by comas and different x and y coordinates are separated by spaces.

Building this out easily in D3.js is not pretty. Because D3.js loves data visualization and pretty things, the D3.js convention is to use the D3.svg.line() generator for polylines and polygons.

To construct the Polyline and Polygon SVG Basic Shapes using D3.js, we will have to learn about **SVG Paths**