

# Line charts

## SVG <path> elements

```
<svg width = "500" height = "400">  
  <path d="M 50 400 L 100 300 L 150 300  
    L 200 330 L 250 175 L 300 275  
    L 350 250 L 400 125" fill="none"  
    stroke="red" stroke-width="5">  
  </path>  
</svg>
```

**d= attribute:**

**M = move to**

**L = line to**

```
<svg width = "500" height = "400">  
  <path d="M 50 400 L 100 300 L 150 300  
    L 200 330 L 250 175 L 300 275  
    L 350 250 L 400 125" fill="none"  
    stroke="red" stroke-width="5">  
    </path>  
</svg>
```



# Data for Line Chart

| Day     | High Temp |
|---------|-----------|
| April 1 | 60        |
| April 2 | 43        |
| April 3 | 43        |
| April 4 | 56        |
| April 5 | 45        |
| April 6 | 62        |
| April 7 | 49        |

# Back to line charts

## What we have:

| Day     | High Temp |
|---------|-----------|
| April 1 | 60        |
| April 2 | 43        |
| April 3 | 43        |
| April 4 | 56        |
| April 5 | 45        |
| April 6 | 62        |
| April 7 | 49        |

## What we need:

```
<path class="line" fill="none"
d="M0,116.66666666666669C27.7777777777
7786,155.5131766381766,55.55555555555557
,
194.35968660968655,83.33333333333333,215
.
83333333333331C111.11111111111109,237.3
0698005698008,138.88888888888889,241.407
76353276354,166.66666666666666,215.83333
333333331C194.44444444444443,190.258903
1339031,222.22222222222223,135.009259259
25924,250,140C277.77777777777777,144.990
74074074076,305.55555555555554,210.22186
609686608,333.3333333333333,204.16666666
666666C361.11111111111111,198.1114672364
6723,388.88888888888889,120.7699430199430
4,416.6666666666667,105.00000000000003C4
44.444444444444446,89.23005698005701,472.
22222222222223,135.03169515669518,500,18
0.833333333333334"></path>
```

# Step 1. Create a line generator

**Expects data in an array of 2-dimensional arrays, that is, an array of (x,y) pairs:**

```
> var dataset = [ [0, 60], [1, 43], [2, 43],  
                  [3, 56], [4, 45], [5, 62],  
                  [6, 49] ];
```

```
> var mylinegen = d3.line()
```

**Test it in the Console:**

```
> mylinegen(dataset);
```

# Add a path element

**Just to see how it works:**

```
> var mypath = mylinegen(dataset);
```

```
> d3.select("svg").append("path")  
  .attr("d", mypath)  
  .attr("fill", "none")  
  .attr("stroke", "red")  
  .attr("stroke-width", "5");
```

# Add scales

**An ordinal scale for x:**

```
> var xScale = d3.scaleBand()  
    .domain(d3.range(dataset.length))  
    .range([0, 500])
```

**A linear scale for y:**

```
> var yScale = d3.scaleLinear()  
    .domain([d3.min(dataset, d => d[1]) - 20,  
            d3.max(dataset, d => d[1]) + 20])  
    .range([400, 0]);
```

# Add accessor functions `.x( )` `.y( )`

```
> mylinegen  
  .x(d => xScale(d[0]))  
  .y(d => yScale(d[1]));
```

**Test again:**

```
> mylinegen(dataset);
```



# Add a path element

**Try the new and improved line generator:**

```
> var mypath = mylinegen(dataset);
```

```
> d3.select("svg").append("path")  
  .attr("d", mypath)  
  .attr("fill", "none")  
  .attr("stroke", "red")  
  .attr("stroke-width", "5");
```

## Step 2. Call the line generator

```
> d3.select("svg")  
  .append("path")  
  .datum(dataset)  
  .attr("d", mylinegen)  
  .attr("fill", "none")  
  .attr("stroke", "teal")  
  .attr("stroke-width", "5");
```

# Or: Create a line style

```
.line {  
    fill: none;  
    stroke: teal;  
    stroke-width: 5px;  
}
```

```
d3.select("svg")  
  .append("path")  
  .datum(dataset)  
  .attr("d", mylinegen)  
  .attr("class", "line");
```

EDAV8\_1\_linegen.html