

Getting started with Plotly.js

In this lesson, we'll learn the basics of drawing charts with Plotly.js

WE'LL COVER THE FOLLOWING ^

- Plotly

Often, in real world applications, there's some third party library that needs to be used, or maybe some legacy code that needs to be bridged. To showcase how one would go about doing that, we'll be using [Plotly](#) to draw a beautiful temperature forecast graph!

Plotly

We'll explore the Plotly.js API and draw a scatter chart.

To use `plotly.js`, we need to add it to our application. Copy and paste the following line into your HTML:

```
<script src="https://cdn.plot.ly/plotly-1.8.0.min.js"></script>
```

Including this script gives us access to the `Plotly` variable in our code. Using `Plotly.newPlot`, we can easily create graphs to showcase the weather data:

```
Plotly.newPlot();
```

Let's look at the `newPlot` function in a little bit more detail:

1. The first argument is the `id` of the DOM element we want to render our plot into.
2. The second argument is an array of objects with a few properties for our plot, with `x`, `y` and `type` being the most relevant for us.

Plotly.js makes it easy to create a wide variety of plots, the one we care about

the most at the moment is a `scatter` plot. See the documentation [here](#) for

some examples.

This is what our `Plotly.newPlot` call will look like:

```
Plotly.newPlot('someDOMElementId', [{
  x: ourXAxisData,
  y: ourYAxisData,
  type: 'scatter'
}], {
  margin: {
    t: 0, r: 0, l: 30
  },
  xaxis: {
    gridcolor: 'transparent'
  }
}, {
  displayModeBar: false
});
```

As you can see, we also pass in some styling information as the third argument (we specify a few margins and hide the xaxis grid lines), and some options as the fourth argument. (we hide the mode bar).

Plotly.js has tons of options, I encourage you to check out the [excellent documentation](#) and play around with a few of them!

For getting our feet wet, we'll draw a chart of 9 numbers [-4, 4], their squares and their cubes. Here's the data that we will use for our chart:

`x-axis` : [-4, -3, -2, -1, 0, 1, 2, 3, 4]

`y-axis-1` : [16, 9, 4, 1, 0, 1, 4, 9, 16];

`y-axis-2` : [-64, -27, -8, -1, 0, 1, 8, 27, 64];

HTML JavaScript Output

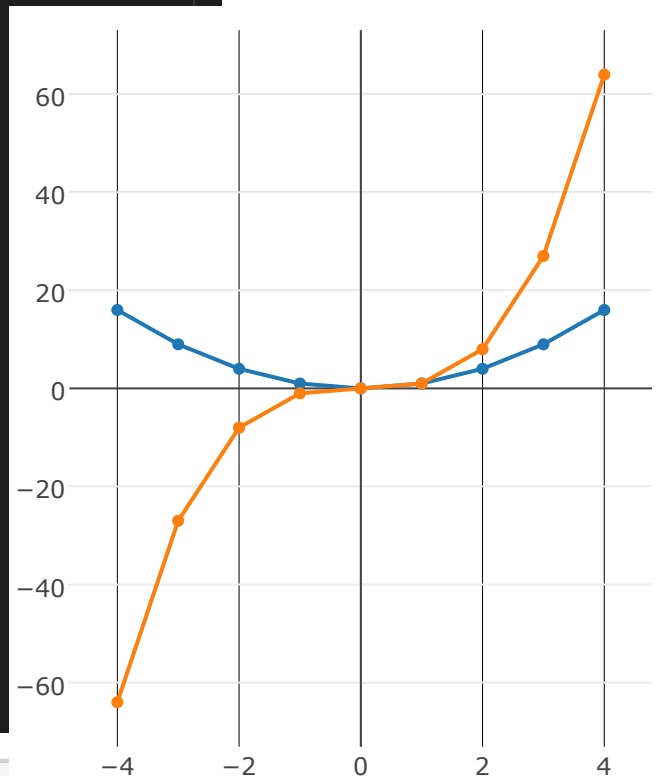
```
1 let numbers = [-4, -3, -2, javascript 1, 2, 3, 4];
2 let squares = [16, 9, 4, 1, 0, 1, 4, 9, 16];
3 let cubes = [-64, -27, -8, -1, 0, 1, 8, 27, 64];
4
5 Plotly.newPlot('content', [{
6   name: 'squares',
7   x: numbers
```

output

```

7   x: numbers,
8   y: squares,
9   type: 'scatter'
10  }, {
11   name: 'cubes',
12   x: numbers,
13   y: cubes,
14   type: 'scatter'
15  }], {
16   margin: {
17     t: 0, r: 0, l: 30
18   },
19   xaxis: {
20     gridcolor: 'transparent'
21   }
22 }, {
23   displayModeBar: false
24 });

```



As you can see, it's perfect for a weather forecast. But we are in the React world where everything is (or at least should be) a **component**. You might have already guessed it - it's time to build our **Plot** component.