tt()

D3 FTW!

Schedule

1. Why you’re using D3
2. D3 Concepts
3. Let’s get started

Schedule

1. **Why you're using D3**
2. D3 Concepts
3. Let's get started

D3

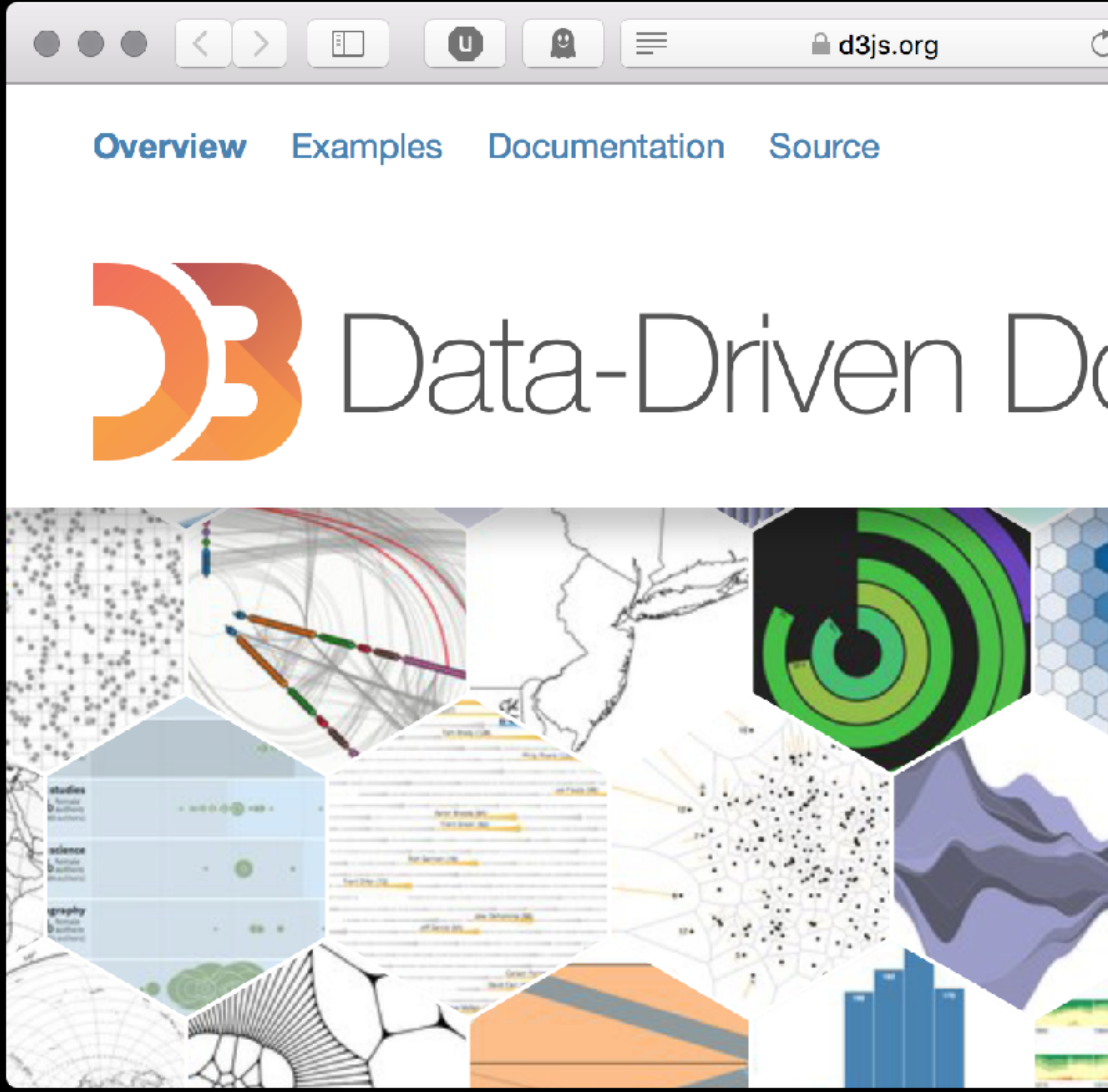
Data **D**riven **D**ocuments

D3 – the purpose

"D3.js is a JavaScript library for **manipulating documents based on data.**

D3 helps you bring data to life using HTML, SVG, and CSS.

(…) D3 is extremely fast, supporting large datasets and dynamic behaviours for interaction and animation."

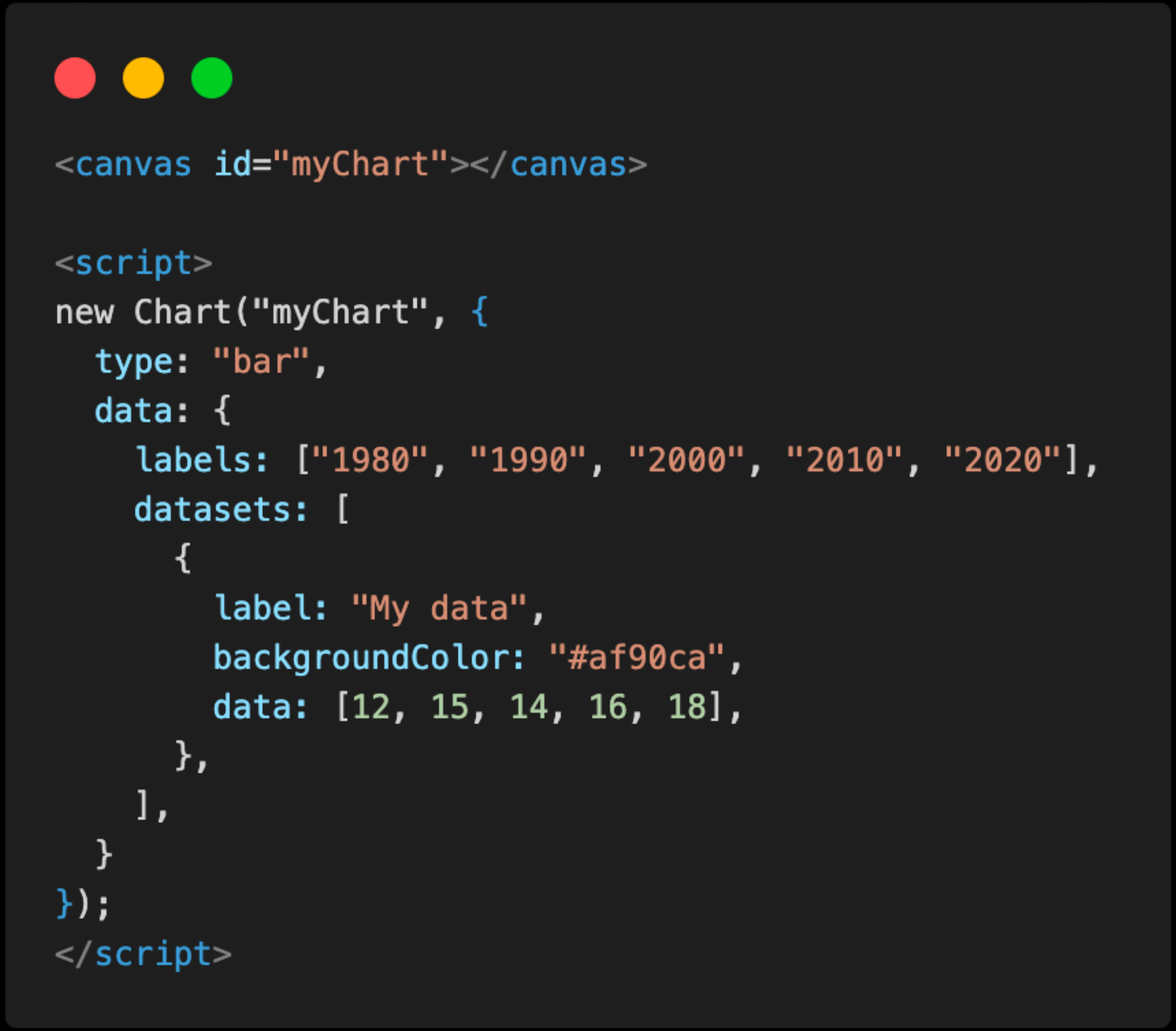


D3 vs Instant Mix

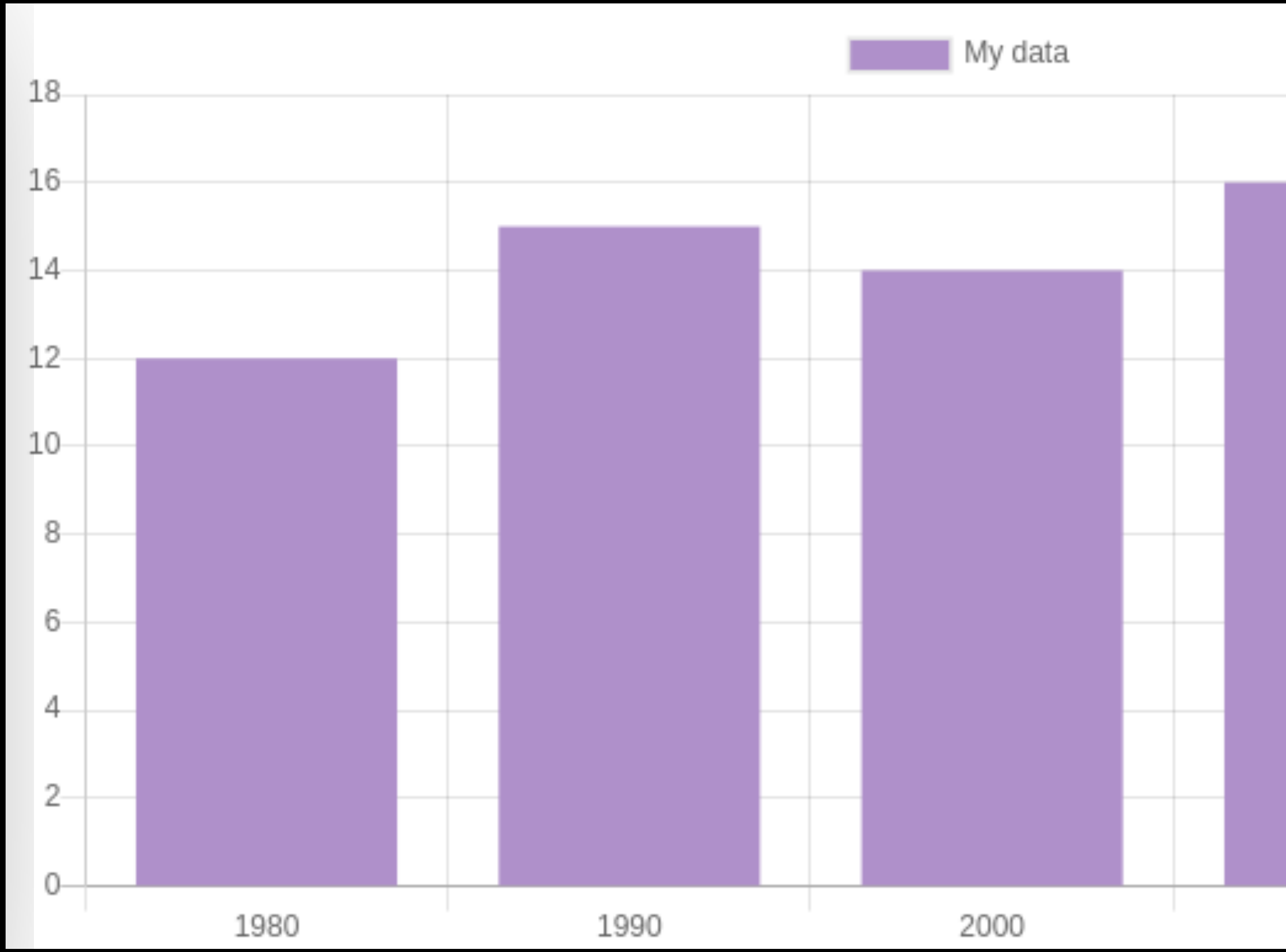
- RawGraphs.io
- LocalFocus.nl
- Flourish.studio

Just add water!

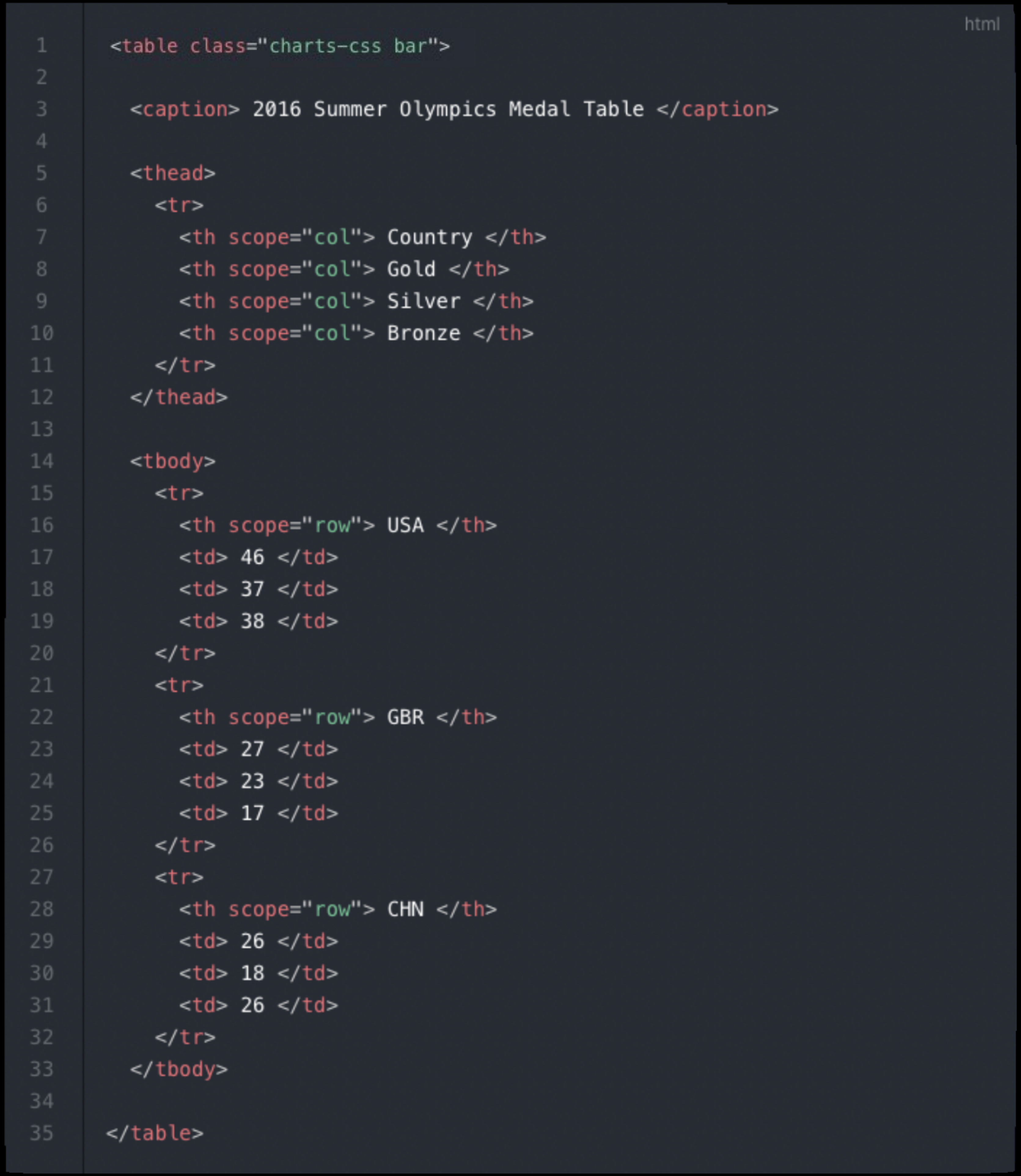
D3 vs Chart.js



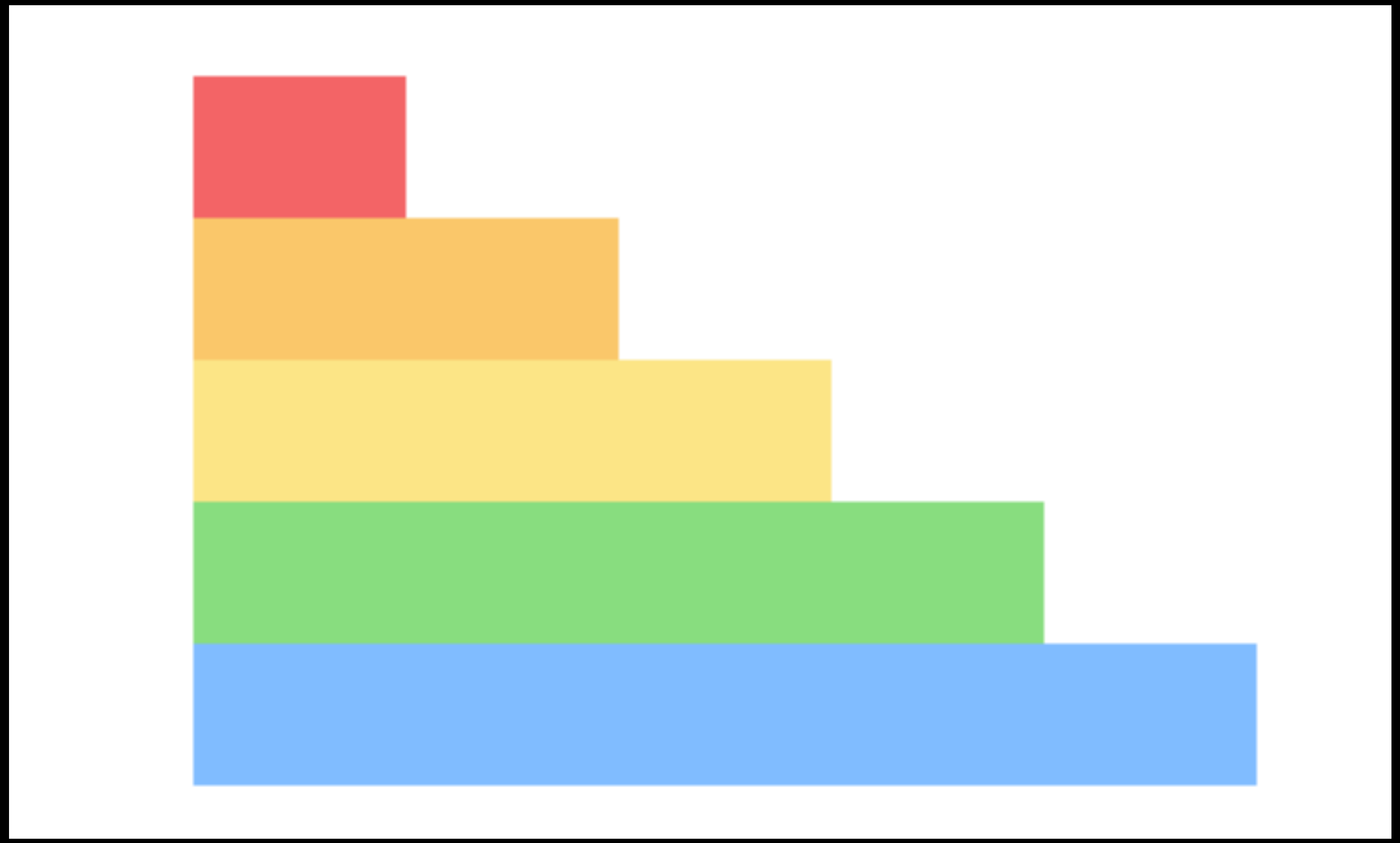
```
<canvas id="myChart"></canvas>

<script>
new Chart("myChart", {
  type: "bar",
  data: {
    labels: ["1980", "1990", "2000", "2010", "2020"],
    datasets: [
      {
        label: "My data",
        backgroundColor: "#af90ca",
        data: [12, 15, 14, 16, 18],
      },
    ],
  }
});
</script>
```



D3 vs Chart.css

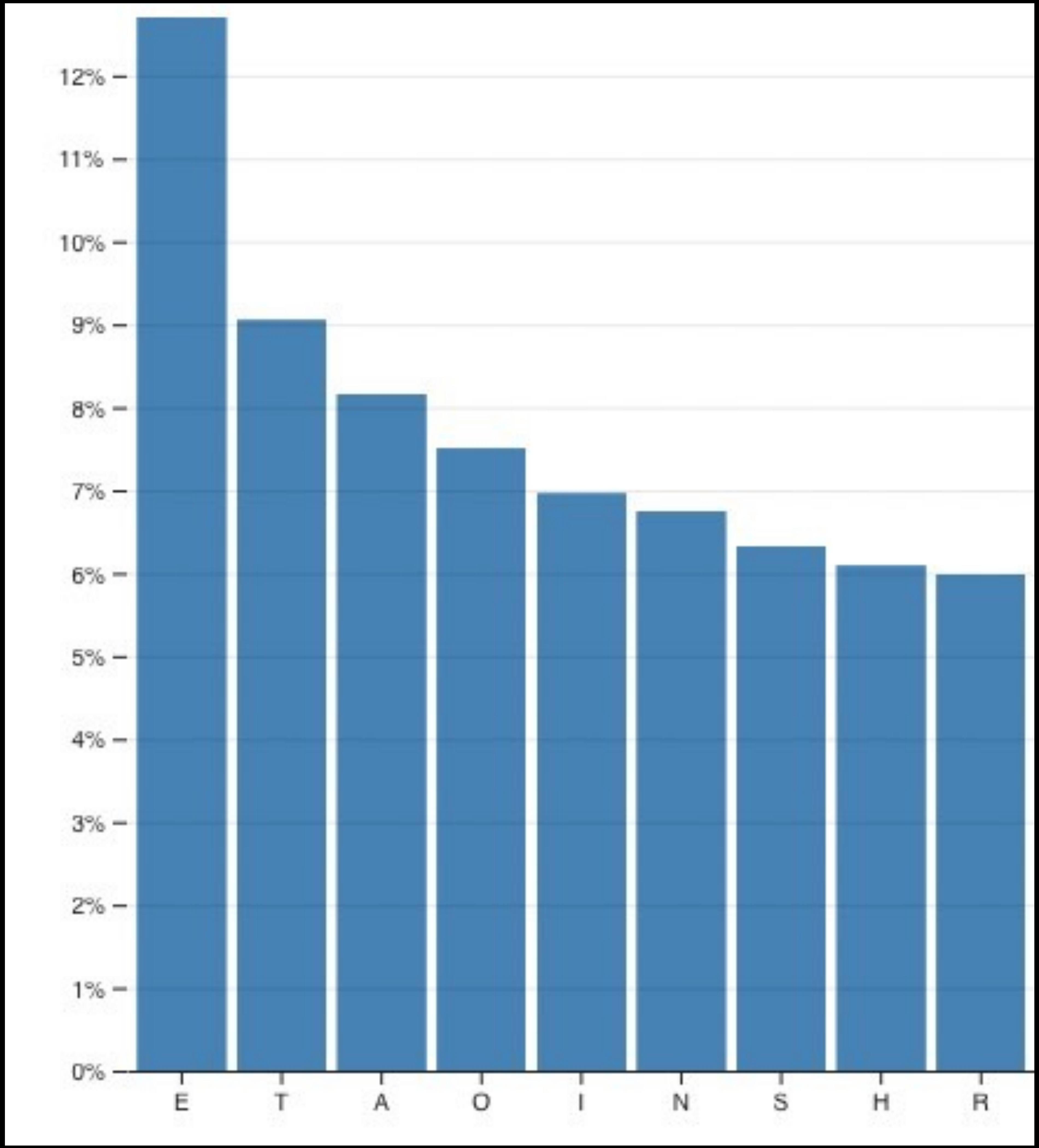


```html
<table class="charts-css bar">

  <caption> 2016 Summer Olympics Medal Table </caption>

  <thead>
    <tr>
      <th scope="col"> Country </th>
      <th scope="col"> Gold </th>
      <th scope="col"> Silver </th>
      <th scope="col"> Bronze </th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <th scope="row"> USA </th>
      <td> 46 </td>
      <td> 37 </td>
      <td> 38 </td>
    </tr>
    <tr>
      <th scope="row"> GBR </th>
      <td> 27 </td>
      <td> 23 </td>
      <td> 17 </td>
    </tr>
    <tr>
      <th scope="row"> CHN </th>
      <td> 26 </td>
      <td> 18 </td>
      <td> 26 </td>
    </tr>
  </tbody>

</table>
```



D3 Concepts

1. Why you're using D3
2. **D3 Concepts**
3. Let's get started



D3 Concepts

1. Selections
2. Joins
3. Accessor functions
4. Scales
5. Axes
6. Ticks

Selections - the purpose

We work with HTML elements, SVG …. and want to modify them

d3.select() is kinda like document.querySelector()

d3.selectAll is kinda like document.querySelectorAll()

```
d3.selectAll('circle')
```

```
d3.selectAll('circle').style('fill', 'red')
```

Selections

d3.select() is kinda like document.querySelector()

d3.selectAll is kinda like document.querySelectorAll()

Name	Behaviour	Example
.style	Update the style	d3.selectAll('circle').style('fill', 'red')
.attr	Update an attribute	d3.selectAll('rect').attr('width', 10)
.classed	Add/remove a class attribute	d3.select('.item').classed('selected', true)
.property	Update an element's property	d3.selectAll(‘input[type=checkbox]’).property('checked', true)
.text	Update the text content	d3.select(‘h1').text('Hello world’)
.html	Change the html content	d3.select(‘form').html('<button>Turn off</button>’)

https://codepen.io/vijnv/pen/ExrNBYL

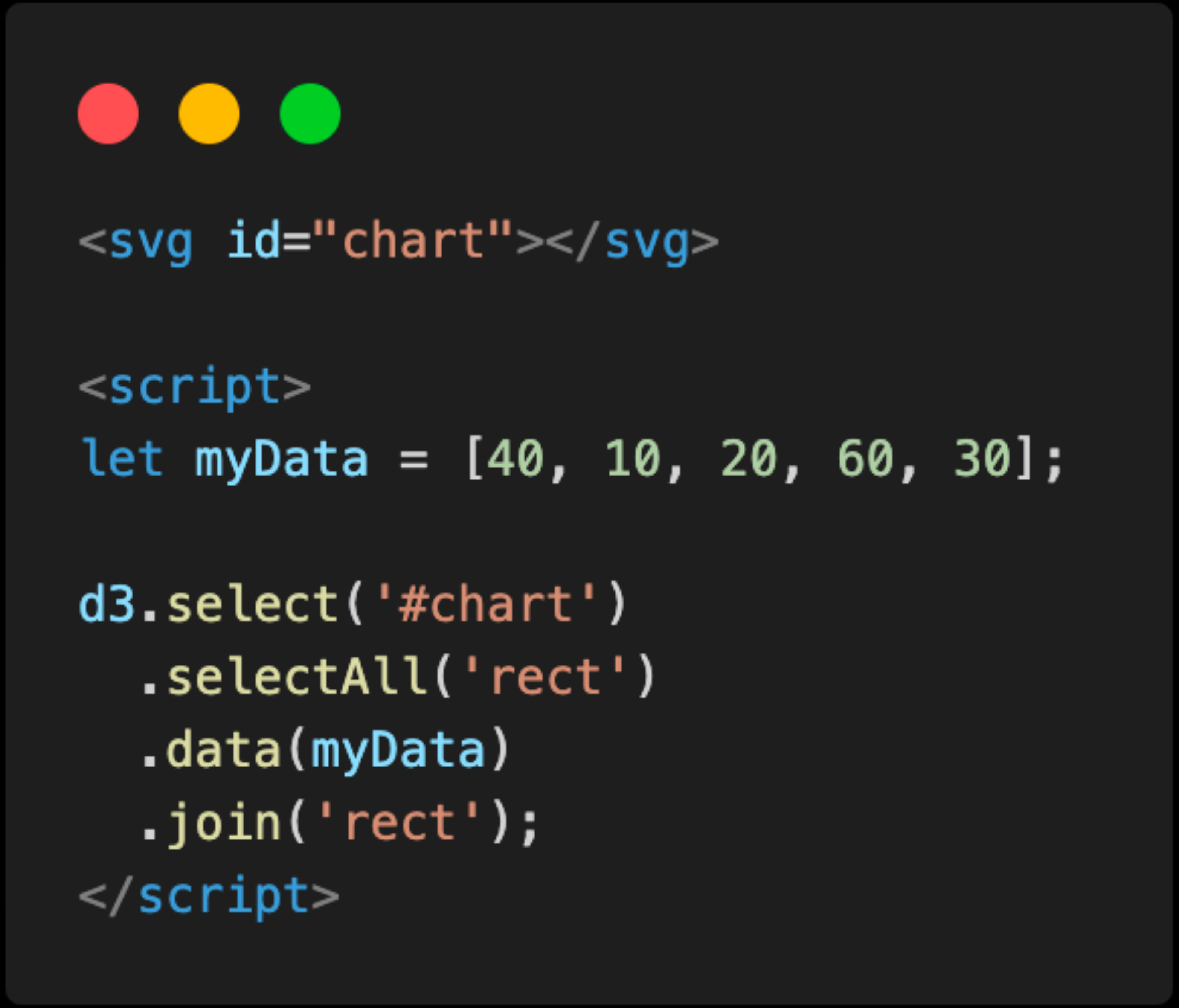
Joins - the purpose

We never know in advance how many SVG elements we will need.

It depends on our data

Joins

Data joins are kinda like doing a mail merge in Office to create address labels based on a list in Excel

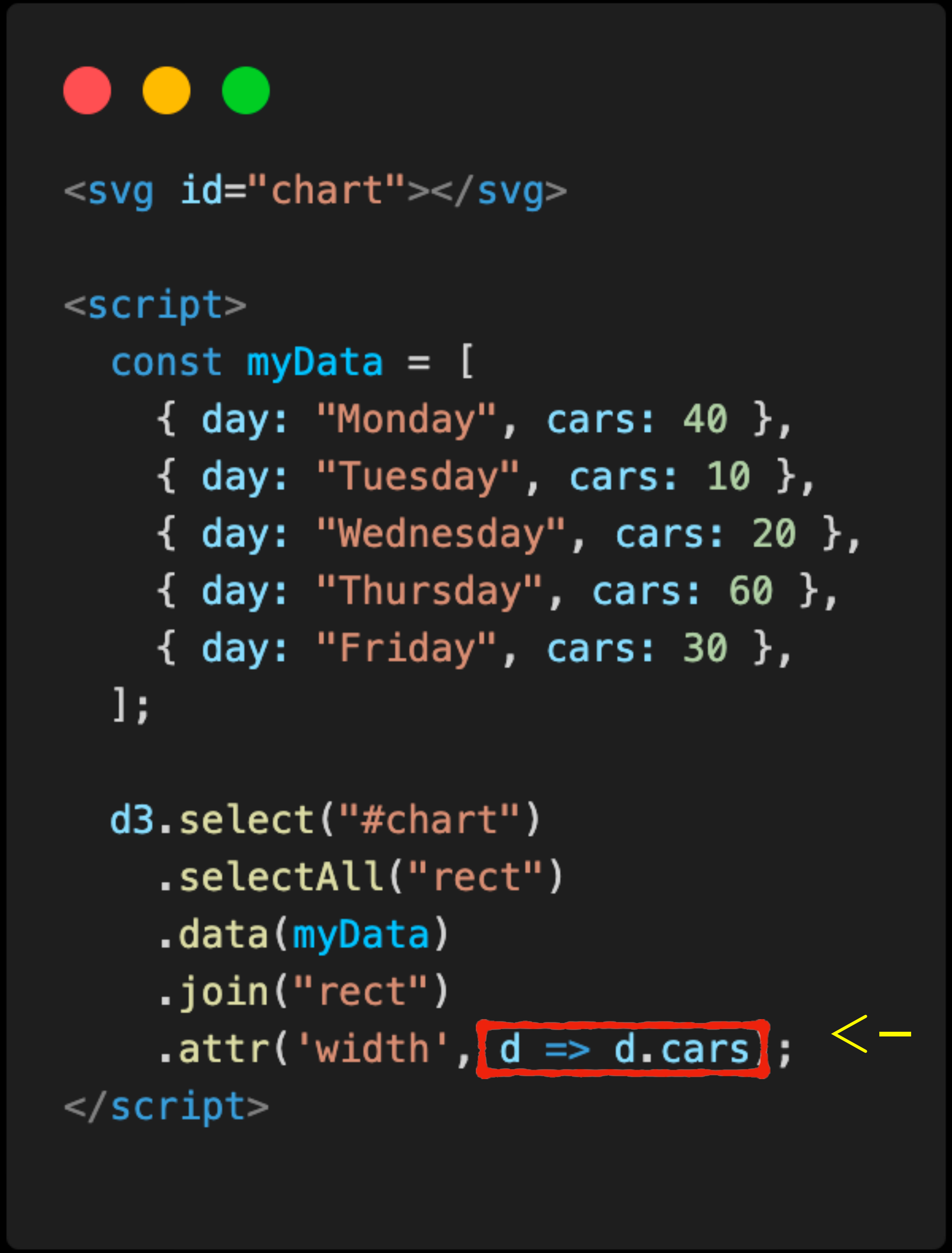


```
<svg id="chart"></svg>

<script>
let myData = [40, 10, 20, 60, 30];

d3.select('#chart')
  .selectAll('rect')
  .data(myData)
  .join('rect');
</script>
```

Here we use d3.join() to create a <rect> element for each item in our myData array

https://codepen.io/vijnv/pen/poGNmme

Accessor functions - the purpose



```
<svg id="chart"></svg>

<script>
  const myData = [
    { day: "Monday", cars: 40 },
    { day: "Tuesday", cars: 10 },
    { day: "Wednesday", cars: 20 },
    { day: "Thursday", cars: 60 },
    { day: "Friday", cars: 30 },
  ];

  d3.select("#chart")
    .selectAll("rect")
    .data(myData)
    .join("rect")
    .attr('width', d => d.cars);
</script>
```

<- Accessor function

If you're using JSON (an array of objects) you'll need to tell D3 which property you want to use

Schedule

1. Why you're using D3
2. D3 Concepts
3. **Let's get started**

Exercise

1. Live demo!
2. Practice with selections, joins and accessor function:
    1. https://codepen.io/Laura_B/pen/KKOBMgL

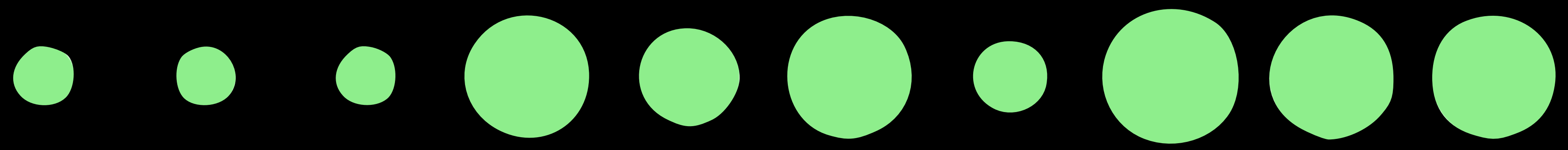
D3 Concepts

1. Why you're using D3
2. **More D3 Concepts**
3. Let's get started

Scales - the purpose

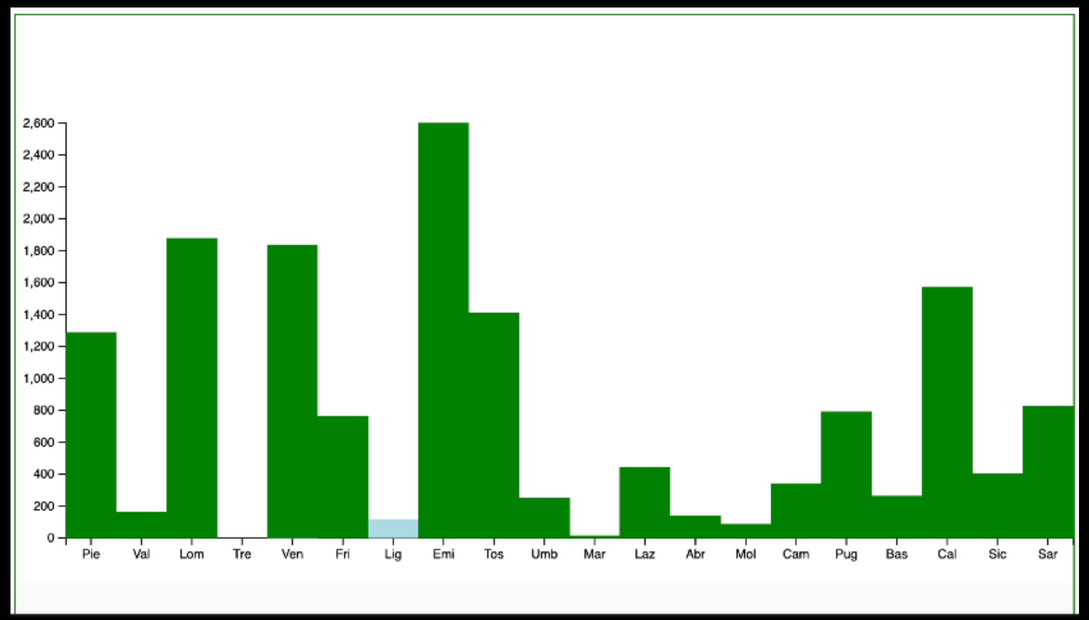
Scales allow to visualize data through:

size



position

(In a way they fit in your graphic view)



Scales - the math

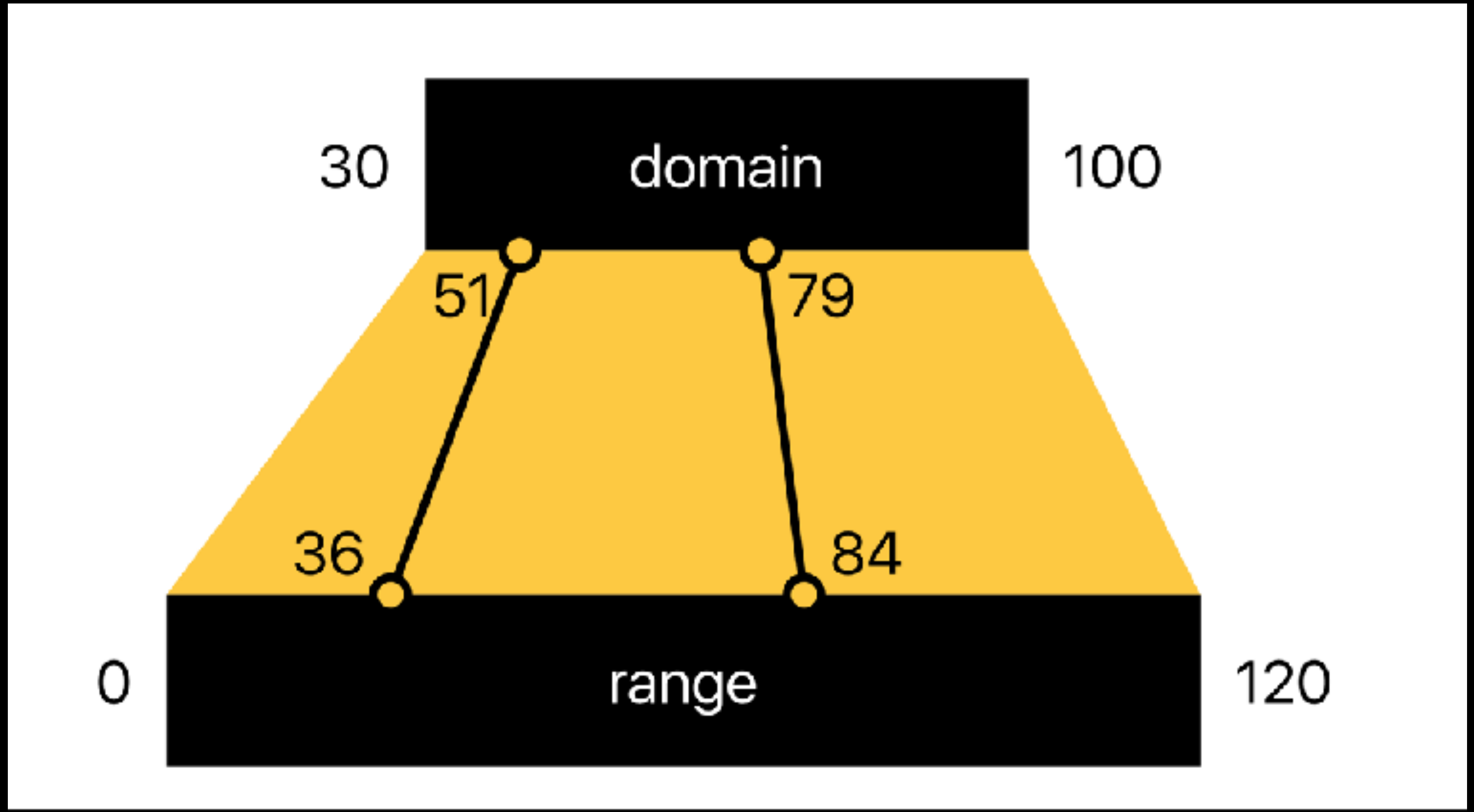
Scales help you calculate

- how big elements of your graph should be
- where are they positioned

They are **functions**

domain = what you need to show

range = the options you have



We will cover this tomorrow

Scales

Scales allow to visualize data through:

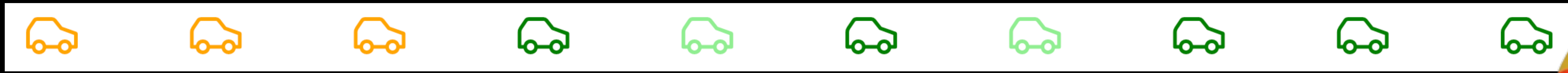
size

position

and **color**

Huiswerk - voor morgen

- Werk in tweetallen
- Maak een **point scale** die er voor zorgt dat, hoeveel ritten er ook worden afgebeeld door een auto, ze altijd passen in een rij van 800px
- Maak een **kleurscale** die de user laat zien hoe rendabel de rit is, bijvoorbeeld:



- **Zie: https://www.d3indepth.com/scales/**

Wij zijn benieuwd!!!!

```
Uncaught SyntaxError
Unexpected end of input
```