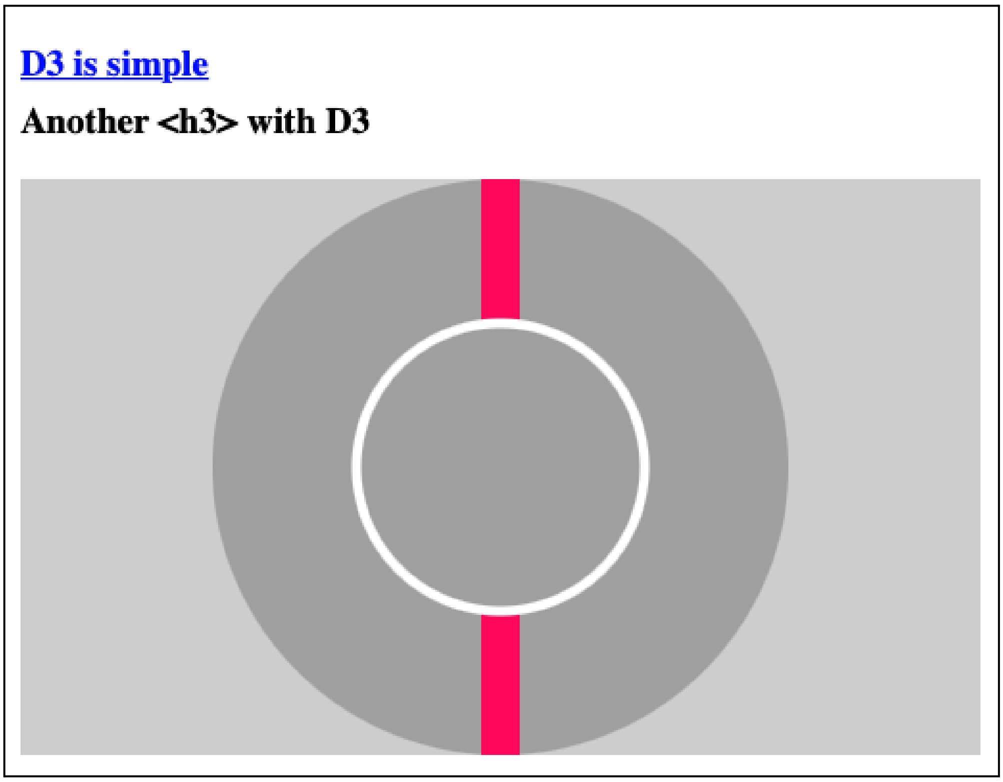
Exercises 2 — D3.js

Important notice! Always save the solutions for the exercises. They will be submitted and evaluated (*Avaliação Contínua*). Confirm the exact date in *Inforestudante*.

Exercise A



D3 is simple

Another <h3> with D3

Figure 1. Webpage screenshot.

Create a new WebStorm/other project and write the following code in an HTML document (i.e., “index.html”):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <script type="text/javascript"
src="https://d3js.org/d3.v6.min.js"></script>
```

```
  </head>

  <body>
    <h3>D3 is simple</h3>
    <script type="text/javascript"
src="script.js"></script>
  </body>
</html>
```

1. Create a Javascript file (i.e., “script.js”) and using D3:
    a. Change the colour of <h3> to blue
    b. Add another <h3> bellow the first one with the sentence: “Another <h3> with D3”
    c. Select all <h3> elements and colour them in black.
        i. Colour in blue only the first one.
    d. Add the following lines to your code. Explain what they do.

```
      d3.select('h3').attr('id', 'first');
      d3.select('#first').style('text-decoration',
'underline');
```

    e. Create an SVG element in D3 and append it to the HTML <body>. This SVG must have 500px width, 300px height, and be coloured in lightgray.
    f. With D3, draw rectangles and circles as pictured in Figure 1.

Exercise B

Write the following in a Javascript file:

```
const width = 500;
const height = 300;
let data = [23, 12, 34, 45, 7];

d3.select('body')
  .append('svg')
  .style('width', width)
  .style('height', height)
  .style('background-color', 'rgb(250, 250, 250)');
```

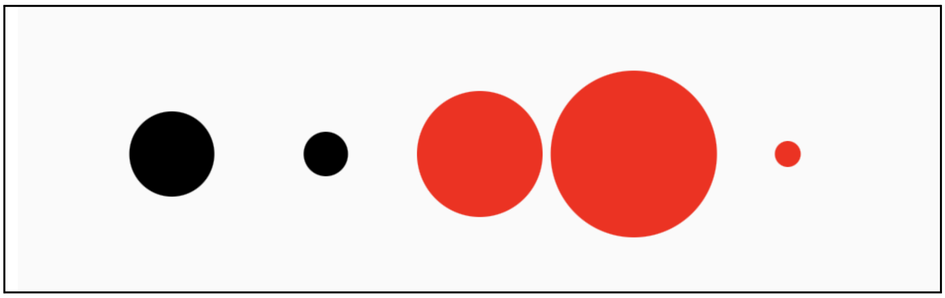
```
let svg = d3.select('svg');

svg.append('circle')
  .attr('cx', 50)
  .attr('cy', height/2)
  .attr('r', 20)
  .attr('fill', 'black');

svg.append('circle')
  .attr('cx', 23)
  .attr('cy', height/2)
  .attr('r', 20)
  .attr('fill', 'black');

// binding data
let circles = svg.selectAll('circle')
  .data(data);
```

1. Update the existing circles, in order to represent the data in the array. If needed, new circles should be added and painted in red. All circles need to fit the SVG and be spaced equally, as in the image below:



   a. Change the array of values (i.e, data) by adding more values to the array. See what happens. What if there is only one value in the array?

      i. Add the following line to your code.

```
circles.exit().remove();
```

      ii. What happens now if the number of array elements is smaller than the number of circles?

Data

The data you are provided with contains approximately 11000 records of fuel prices of the period between 2018 and 2019. The dataset can be accessed through the following link precos_combustiveis_2018_2019.csv. In the WebStorm project, the dataset should be placed in the data subdirectory.

Exercise C

Add to the HTML document a container and give it an id. In the JavaScript file write a callback function which will be used as a response to the completion of the data loading:

```
function functionName(table){
    // code to execute after the data is loaded
}

d3.csv(url).then(functionName);
```

1. Create a SVG element inside the container previously specified and attribute the result to a variable.
    a. Analyse the tree of the elements in the Developer Tool window and find the created SVG element;
    b. Add a "class" attribute to the SVG. Analyse the result;
    c. Add width and height attributes;
2. Create a G element inside the previously created SVG and attribute it to a variable. Analyse the resulting tree of elements.
3. Make a simple data join using circles as a container of the data defined above:
    a. Map the column price to the radius of the circles, multiplying it by 10;
    b. Map the price column to the x position of the circles, multiplying it by, for instance, 200 (note, you should use *cx* and *cy* to position circles)
    c. Colour the circles differently according to the price value: circles in blue, if the price value is lower than 1, or in red, if the price value is equal or higher than 1.