1. How can D3 access and change the DOM? What do select and selectAll do?

As D3 is a JavaScript library it is basically placed on top of JavaScript and therefore uses JavaScript to access the DOM. select and selectAll are helper functions in D3 that allow you to select DOM-elements in an easier way than with pure JavaScript.

1. What are the d and i in function(d){} and function(d, i){}?

In these functions d stands for the data point value, e.g. the value of a bar in a given bar diagram, whilst i stands for the index of that particular data point.

1. Write sample lines of JavaScript to add a div element with class “barChart1” and to add an svg element with class “barChart2” with square dimensions.

In simple JavaScript this would be something like the following:

var div = document.createElement(“div”);

div.className = “barChart1”;

var svg = document.createElementNS("http://www.w3.org/2000/svg", "svg");

svg.setAttribute(“width”, “100px”)

svg.setAttribute(“heigth”, “100px”)

svg.className = “barChart2”

div.appendChild(svg);

var element = document.getElementsByTagName(“body”)[0];

element.appendChild(div);

1. Describe append, update, enter, and exit at a high level. What does “selectAll + data + enter + append” refer to?

append allows you to create and attach an element all at once as long as there is already a selection to attach it to.

update; when your data values change or when some disappear and some are new you want to update your visualisation; or at least you want to update the elements that will stay on stage by re-binding them with the new data.

enter; in combination with a dataset, enter() prepares missing elements to be added, e.g. when there are more data points in the dataset then there are bars in the bar graph. So, this helps you create new svg elements to show missing data.

The sequence of methods described in this question allows you to create elements based on your data (and the elements that are still missing) and subsequently append them to your DOM.

exit; allows you to remove svg elements from your visualisation when those data points are no longer relevant to show.

1. What are the main differences between drawing a bar chart with HTML and SVG?

In HTML (+CSS) you can make a bar diagram by creating a group of div’s, each with a specific width corresponding to the value. By setting the background colour, you create a rectangle of the specified width. Of course, this is hard coding the graph in your DOM.

You’ll still use a separate style sheet, but with SVG the semantics are somewhat different (fill instead of background-color) and there are more options for shapes. Also, the individual bars are not in div elements but in g’s and text and the rectangle shape needs to be added in separate tags inside the g elements. Furthermore, the elements must now be absolutely positioned; i.e. translations must be made in order to be able to hard-code the positions relative to the origin of the graph.

1. In drawing the simple bar chart with D3 and SVG, what elements were appended, and to what parts of the graph did these elements correspond?

In the simplest of bar charts you need rectangles to represent the values of your data; these rectangles are therefore the bars in the bar graph. These bars are represented with g elements in which rectangle elements and text elements are positioned. This results in the same SVG code as would be created when drawing the graph entirely in SVG. However, instead of writing the same code for each bar, in D3 you can use the selectAll() method and draw a rectangle for each data point in your dataset at once.