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

Met select kan je 1 “object” selecteren en hiervoor bepaalde attributen toeschrijven. Bij selectAll() worden de attributen voor alle aan te maken objecten aanschreven. Via d3 is het mogelijk om Dom elementen te selecteren en hier waardes / atttributen aan toe te voegen. Of de data uit deze DOM elementen te halen.

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

“d” is een data element dat in een javascript functie gaat die iets returnt. I is de index vand dat element.

3. 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.

```
var barChart1 = d3.select(".barChart1")

    .attr("width", width + margin.left + margin.right)

    .attr("height", height + margin.top + margin.bottom)

   .append("g")

          .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

d3.select(“body”)

          .append(“svg)

                .attr(“class”, “barChart2”)

                .attr(“width”, width)

                .attr(“height”, width)
```

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

append a new element as the last child of the element in the current selection.

because of enter() and exit it is possible to change the data dynamically.

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

SVG is an earlier standard for drawing shapes in browsers. However, SVG is at a fundamentally higher level because each drawn shape is remembered as an object in a scene graph or DOM, which is subsequently rendered to a bit map. This means that if attributes of an SVG object are changed, the browser can automatically re-render the scene.

In the example above, once the rectangle is drawn, the fact that it was drawn is forgotten by the system. If its position were to be changed, the entire scene would need to be redrawn, including any objects that might have been covered by the rectangle. But in the equivalent SVG case, one could simply change the position attributes of the rectangle and the browser would determine how to repaint it. It is also possible to paint a canvas in layers and then recreate specific layers.

SVG images are represented in XML, and complex scenes can be created and maintained with XML editing tools.

The SVG scene graph enables event handlers to be associated with objects, so a rectangle may respond to an onClick event. To get the same functionality with canvas, one must manually match the coordinates of the mouse click with the coordinates of the drawn rectangle to determine whether it was clicked.

Conceptually, canvas is a lower level protocol upon which SVG might be built.[citation needed] However, this is not (normally) the case—they are independent standards. The situation is complicated because there are scene graph libraries for Canvas, and SVG has some bit map manipulation functionality.

http://stackoverflow.com/questions/4996374/what-is-the-difference-between-svg-and-html5-canvas

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

In this case “g” to your dataset.