Problem 1: Readings on D3 and SVG – Bart Quaink - 1121424

1. D3 accesses the DOM via certain commands built into the D3 library. The DOM is created in the HTML page and the D3 library than selects it via the .select function. A variable can be created by using this function, then modified using the D3 library. By appending a certain dataset, e.g. (“svg:svg”) for an svg, a canvas and then giving the visualization certain attributes creates a visualization on the webpage. Different kinds of attributes can be given to the canvas element, like width, height, x and y location and colour.
   Selecting the DOM can be done using the .select and .selectAll functions. Select preserves the original grouping of the DOM data on the HTML page, whereas selectAll creates a new grouping.

2. Often, functions are being written with an input argument of d. D is nothing but convention, it could be called anything. D is a standard fare in d3 code because of M. Bostock’s writing style. The I in the extended function is for representing the order of the elements in the selection. So d actually represents the value of the data, and i represents the value of the index in the dataset.
3. Div elements can be created on this way:

   ```
   var div = document.createElement(“barChart1”);

   div.style.width = “50px”;

   div.style.height=”50px”;

   document.body.appendChild(“barChart1”);
   ```

   Svg elements can be created in JavaScript in a quick and easy way with d3, an example:

   ```
   var container = d3.select(“body”).append(“svg”)
                                        .attr(“width”, 200)
                                        .attr(“height”, 200);
   var square = container.append(“.barChart2”)
                                        .attr(“x”, 25)
                                        .attr(“y”, 25)
                                        .attr(“width”, 100)
                                        .attr(“height”, 100);
   ```

4. *Append* creates as many elements as there are placeholders returned by enter. When appending items they need to be created somewhere, to be able to build upon them a container.

   Update basically is applying the data operator to the selection. Joining data items and DOM elements is updating the data. For data items that aren’t visible in the structure yet, added by the enter selection, update creates this element.

Enter prepares a new element for every unmatched data item. It returns the enter selection, the placeholder nodes for each data element for which no corresponding DOM element exists. No corresponding key in the data means it will become the enter selection.

Exit returns the exit selection. The exit selection are the existing DOM elements in the current selection for which no new data element was found. Where there is no corresponding key in the data will become the exit selection. It merely returns a reference to the exit selection, and does not actually remove the nodes. This is up to the programmer to actually remove them using the *remove* operator.

selectAll + data + enter + exit is known for being the easiest and proper way of adding new elements. SelectAll selects all the elements, data then adds a data set to this set of elements, enter then adds a selection of items to the data set, even though none are drawn yet. These can be updated by using the update selector. Exit does exactly the opposite thing of enter. Instead of waiting for elements to be added, the exit selection contains the data items to be removed from the data set.

5. As SVG stands for scalable vector graphics, it relies on manipulating vector data on a html page. The html canvas does not rely on vector-based graphics, but manipulates the pixels on a page. SVG can be loaded as whole image in XML, which makes it appended to the DOM structure of a page and can then easily be manipulated using CSS and JavaScript. This makes it possible to add event listeners to the SVG element, whereas the html canvas is a simple graphics API. A bar graph can be altered by clicking events of the user, which makes the visualization an interactive piece.

6. Starting off before appending anything data, an empty page that contains a div for the bar chart. This container for the chart has to be selected using D3, and then in that container a child div for each bar is being appended, with the desired with added as a function. Then, using the *data* function, the data of the length of the bar charts is being joined, and afterwards updated. Using enter().append this update is appended to the bar chart.