**F21DV Lab 1**

**Progress:**

Set up GitHub repository configured to use GitHub Pages in order to host the lab exercises directly on the repository.

# Part 2 D3 Setup

## Exercise 1.

The version number displayed in in the console output window is “d3.version: 7.3.0”

A screenshot of a computer

Description automatically generated

## Exercise 2.

Changing different properties of the paragraphs using different selectors:

* select(“p”)
* select(“p1”) - using ID
* selectAll(“p”) - style all paragraphs

Text

Description automatically generatedGraphical user interface, text, application, Word

Description automatically generated

**Code**

**Result**

## Exercise 3.

For this exercise I created a for loop that would iterate 10 times. Inside this loop I created a placeholder variable that would take the value of the new div during each iteration. There is then an if condition that checks if the iterator “i” is below 6; if this is true the div is formatted accordingly. Different formatting is then applied if the condition (i < 6) is not true. I also give each div a class of either “firstFive” or “secondFive” depending on the iteration they are created in so that they can be referenced later if need be.

Text

Description automatically generated

**Code**

**Result**

Graphical user interface, text, application, Word

Description automatically generated

## Exercise 4.

For this exercise I opted to use just the select(“div”) selector as that will select the first div in the list as per the exercise instructions.

Text

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Code**

**Result**

## Exercise 5.

A picture containing table

Description automatically generated

**Result**

**Code**

# Part 3. Data

## Exercise 6.3

I chose to make the value of the new ‘colour’ variable to be ‘blue’ and printed this value in the console along with the other values.

A screenshot of a computer

Description automatically generatedText

Description automatically generated

**Code**

**Result**

## Exercise 7.

I carried out this exercise on the divs created in Part 2, replacing the text in the divs with new values where appropriate.

Text

Description automatically generated

**Result**

**Code**

A picture containing table

Description automatically generated

# Part 4. Data Binding

## Exercise 8.

When starting Part 4 I chose to start a new file to keep each Part logically separate. I completed this exercise by using the typeof operator to check if a value is a number or string.

**Result**

**Code**

Chart, scatter chart

Description automatically generatedText

Description automatically generated

# Part 6. SVG

## Exercise 11.

Initially I found it confusing to align the x1,x2 and y1 and y2 coordinates by after moving each around and c=seeing the result I understood how to position the coordinates in order to create the square.

Shape

Description automatically generated with low confidenceText

Description automatically generated

**Result**

**Code**

## Exercise 12.

At first I had trouble getting d3 to read in the data from a csv locally as I was running the html files off my local machine, however after doing some research I found out that Chrome could not access these local files and so instead had to run a local server. This allowed Chrome to view the local csv and subsequently read in the data.

To run the local server I used the command “python -m http.server” on the command line in the directory where I saved my CSV files. Solution found on stackoverflow: <https://stackoverflow.com/questions/17214293/importing-local-json-file-using-d3-json-does-not-work>

Icon

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

**Result**

**Code**

**Code**

**Result**

## Exercise 13.

Icon

Description automatically generatedText

Description automatically generatedFor this exercise I used the same code structure as before by adding an SVG to the screen and then using a foreach loop to loop through the data and appending each shape with the necessary attributes. However, this time I added the append and attribute function calls inside a join and then enter function. I also added an exit in the join which print and exit message to the console and removes the shape.

**Result**

**Code**