School of Software and Electrical Engineering

COS30045 Data Visualisation

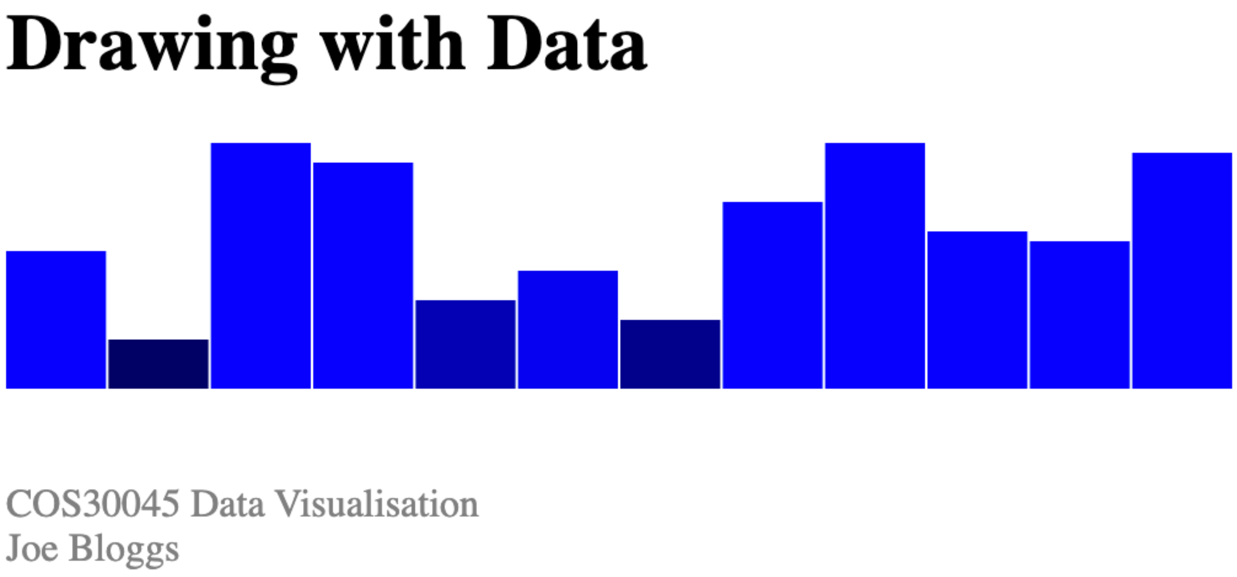
Exercise 2.4 D3 Loading data from CSV file

ILO	Create web-based interactive visualisations using real-world data sets.
Aim:	Use D3 to populate the data set for a bar chart from a CSV file
Resources:	*Textbook:* Chapter 5 Data CSV - Murray (2017) Interactive Data Visualisation (2nd Ed) on ProQuest
Demonstration	If you are required to demonstrate this exercise we will be looking for: - code that is appropriate for exercise, well formatted and commented - code that runs correctly and meets the requirements specified in this exercise - an explain programming features and concepts in the code - the ability to successfully edit code to change a specified feature of the program

Note: This Exercise Guide is not meant to be fully explanatory. You may also need to work through the examples in the text book *Interactive Data Visualisation* for the Web by Murray and/or search the internet.

Overview

In this tutorial we will start using D3 to draw a bar charts. At the end of this exercise you should end up with a bar chart drawn using D3 generated SVGs that looks something like this, but populated from an CSV file instead of hard coding it in the program:



Drawing with Data

COS30045 Data Visualisation
Joe Bloggs

Requirements

□ webpage that displays a bar chart generated from data loaded from a CSV file

□ chart is displayed above footer

□ chart is customised with fill colour that is dependent on the data

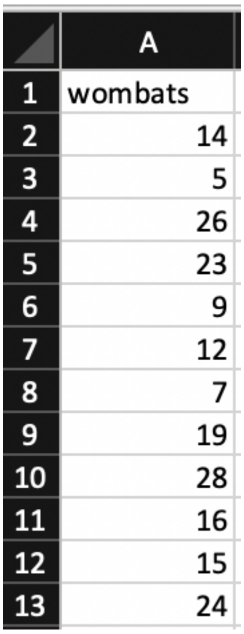
□ html and js code is located in separate files

Step 1: Start a basic HTML template with D3

Start with your code from Drawing with Data Bar Chart. Rename the file and update the meta data and title to reflect this new task.

Step 2: Create a CSV file to read your data from

Open excel and put in one column of data (you can use your data from the Drawing with Data Bar Chart exercise). Make sure you give your column of data a heading (e.g., wombats). D3 expects a heading. Save your data as an CSV file with a meaningful name (e.g., Task_2.4_data.csv). Remember to save as a CSV file (Excel automatically saves in .xlsx format).



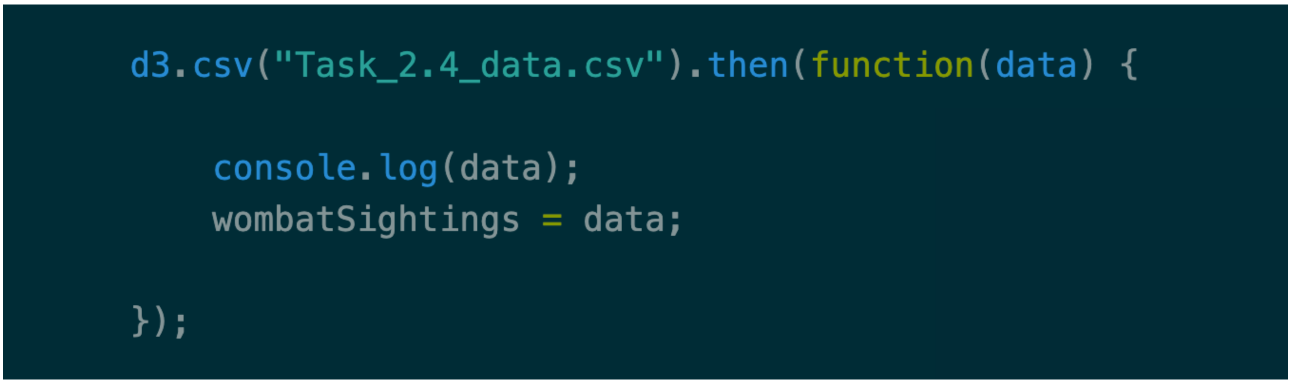
	A
1	wombats
2	14
3	5
4	26
5	23
6	9
7	12
8	7
9	19
10	28
11	16
12	15
13	24

Step 3: Setting up the data

Remove the hard coded data set from your code (i.e., `var dataset = [14. 5. ….]`) as we will now be reading in from the CSV file you just created.

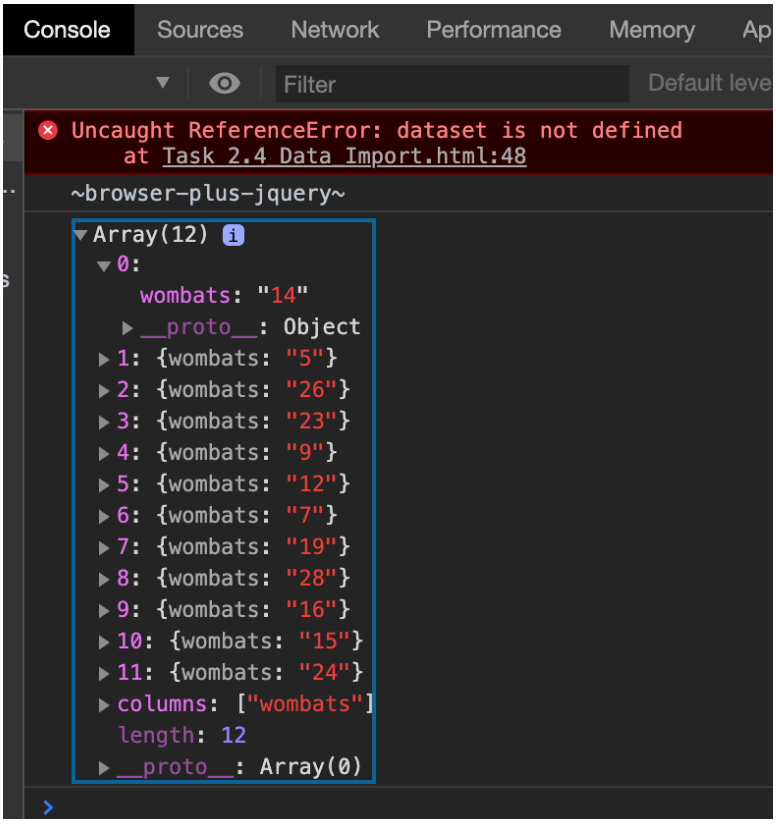
Step 4: Reading in the data

To read in the data we use the d3 function `csv`



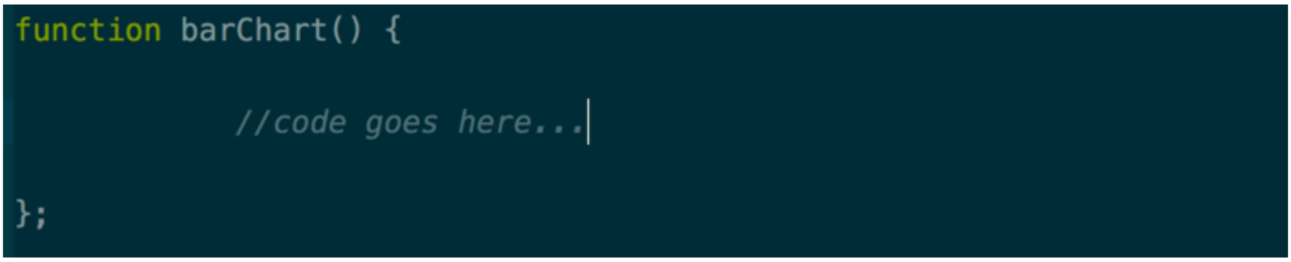
```
d3.csv("Task_2.4_data.csv").then(function(data) {

    console.log(data);
    wombatSightings = data;

});
```

Review the console log to demonstrate the the data is loaded in. At the moment the bar chart won’t work because it doesn’t have access to the data.

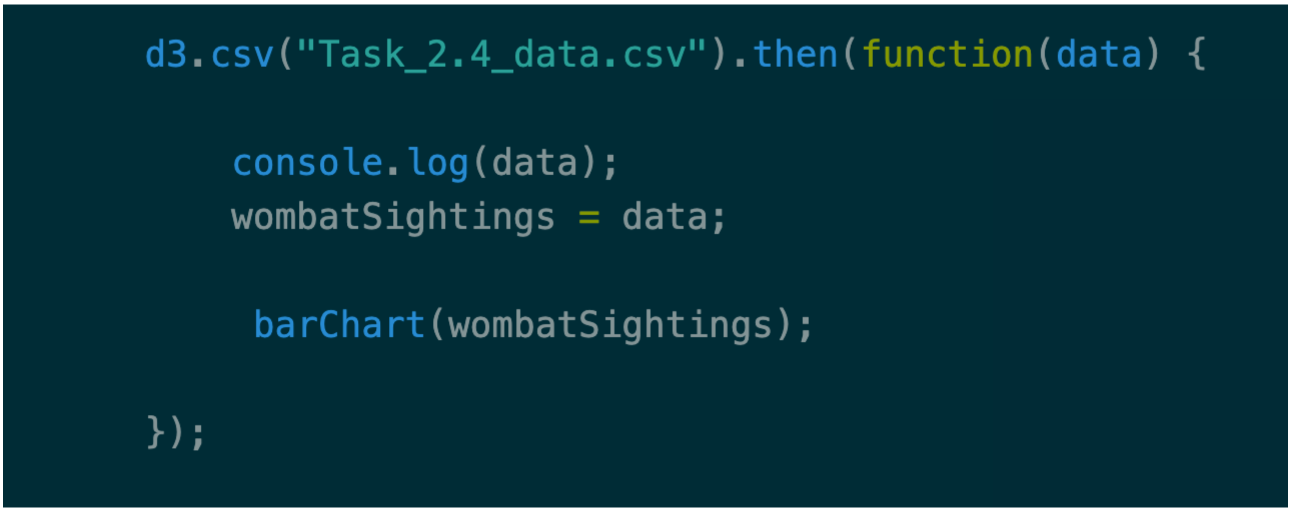


```
Uncaught ReferenceError: dataset is not defined
    at Task 2.4 Data Import.html:48
~browser-plus-jquery~
▼Array(12)
 ▼0:
    wombats: "14"
   ▶__proto__: Object
 ▶1: {wombats: "5"}
 ▶2: {wombats: "26"}
 ▶3: {wombats: "23"}
 ▶4: {wombats: "9"}
 ▶5: {wombats: "12"}
 ▶6: {wombats: "7"}
 ▶7: {wombats: "19"}
 ▶8: {wombats: "28"}
 ▶9: {wombats: "16"}
 ▶10: {wombats: "15"}
 ▶11: {wombats: "24"}
 ▶columns: ["wombats"]
  length: 12
 ▶__proto__: Array(0)
>
```

One way to give our chart code access to the data would be to cut and paste our chart code into the `d3.csv` function. However, that would reduce our ability to reuse the code, so instead we will build a separate bar chart function and call it from within the `d3.csv` function.

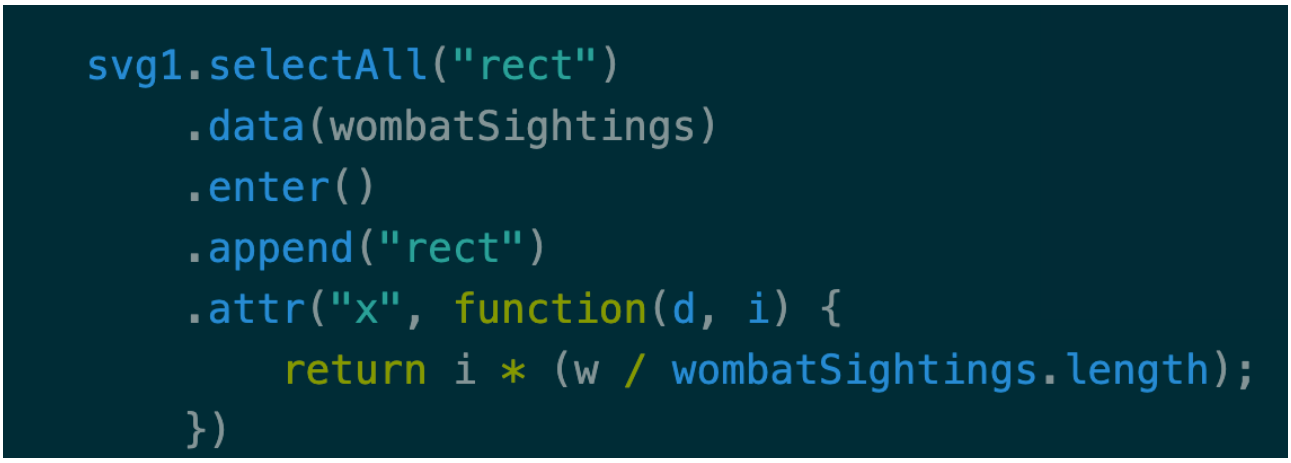


```
function barChart() {

        //code goes here...

};
```



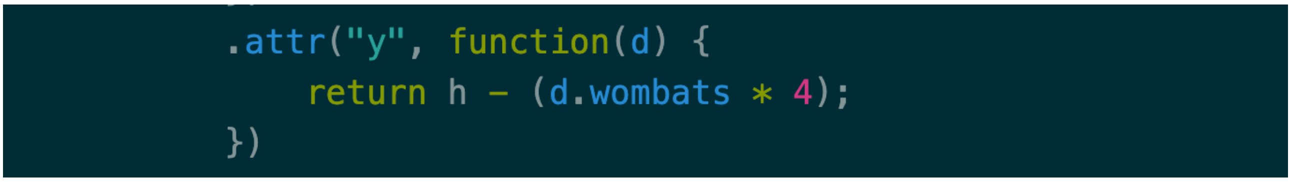
```
    d3.csv("Task_2.4_data.csv").then(function(data) {

        console.log(data);
        wombatSightings = data;

         barChart(wombatSightings);

    });
```

Cut and paste your chart code into the new function. If you changed the name of the data set (like done here (i.e., from 'dataset' to 'wombatSightings')), you will need to update your code so it calls this new data set.



```
svg1.selectAll("rect")
    .data(wombatSightings)
    .enter()
    .append("rect")
    .attr("x", function(d, i) {
        return i * (w / wombatSightings.length);
    })
```

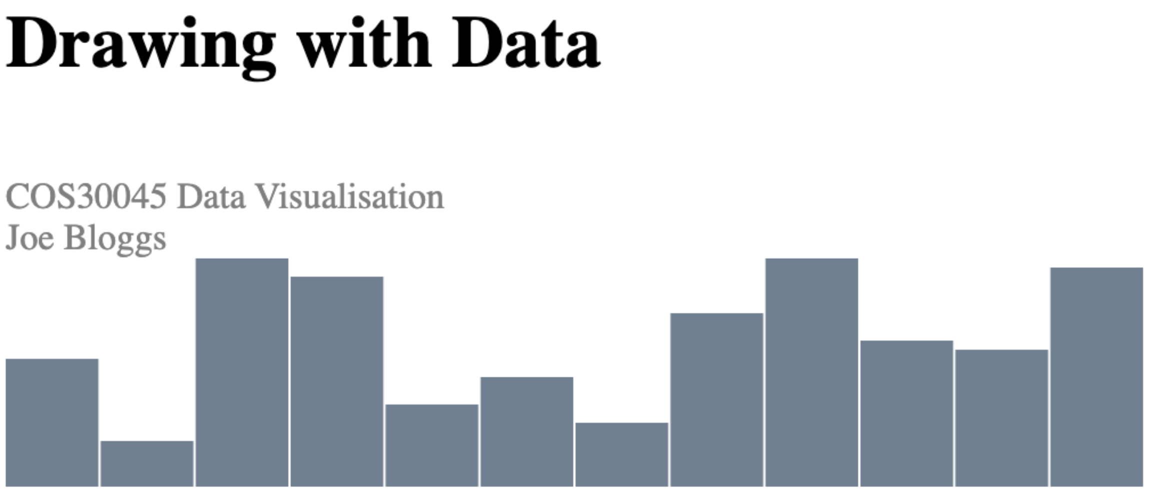
Finally, you need to alter the way you reference the data points to tell D3 which column to get the data from. To access data from a particularly column you use `d.columnname`



```
    .attr("y", function(d) {
        return h - (d.wombats * 4);
    })
```

Update all references to d in your code using the column heading and now your code should produce a chart that uses the imported data.

Tip: This is case sensitive, in this case, `d.Wombats` will not work because the column is written as 'wombats'.



Drawing with Data

COS30045 Data Visualisation
Joe Bloggs

Step 5: Refining the presentation

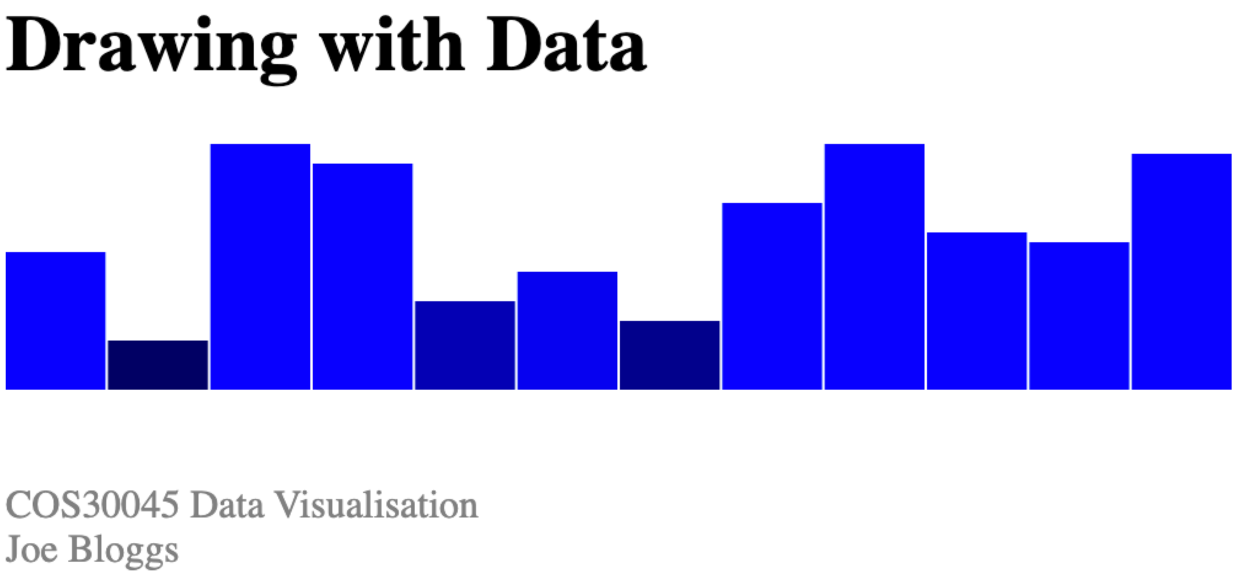
Unfortunately, the chart is not in the right place (i.e., it is below the footer). We can place the chart where we want by using

```
<p id ="chart"></p>
```

and getting D3 to select "#chart" instead of "body".

Fix so that the chart displays above the footer.

Make it so the colour of the bar changes depending on the value of the data:



Drawing with Data

COS30045 Data Visualisation
Joe Bloggs

Finally, if you have done Web Application Design you will know that it is good practice to contain your js script to a separate file.

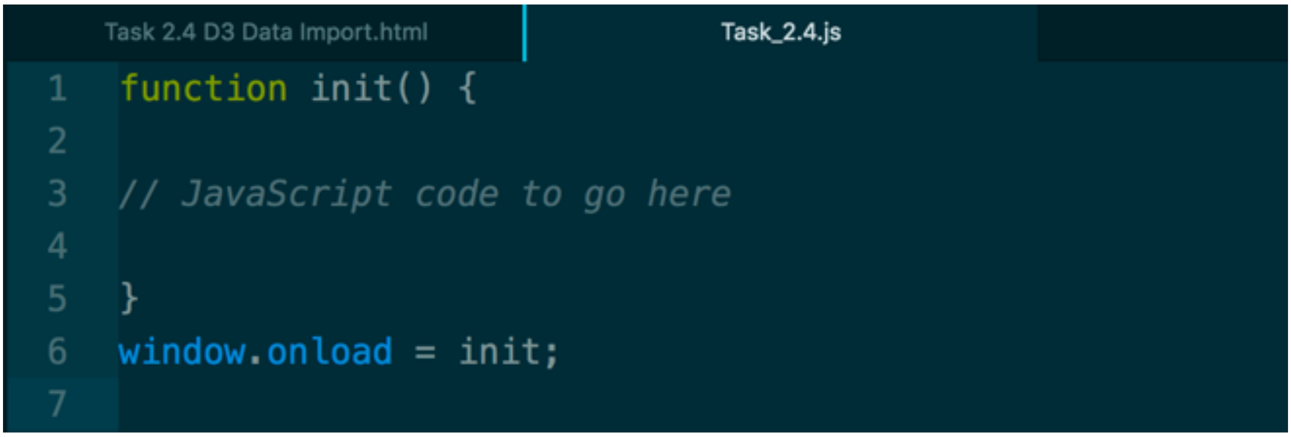
Make sure your D3 script in a separate .js file and you call it in your html file.

We will be using this convention for the rest of our tasks.

Step 6: Creating separate JavaScript files

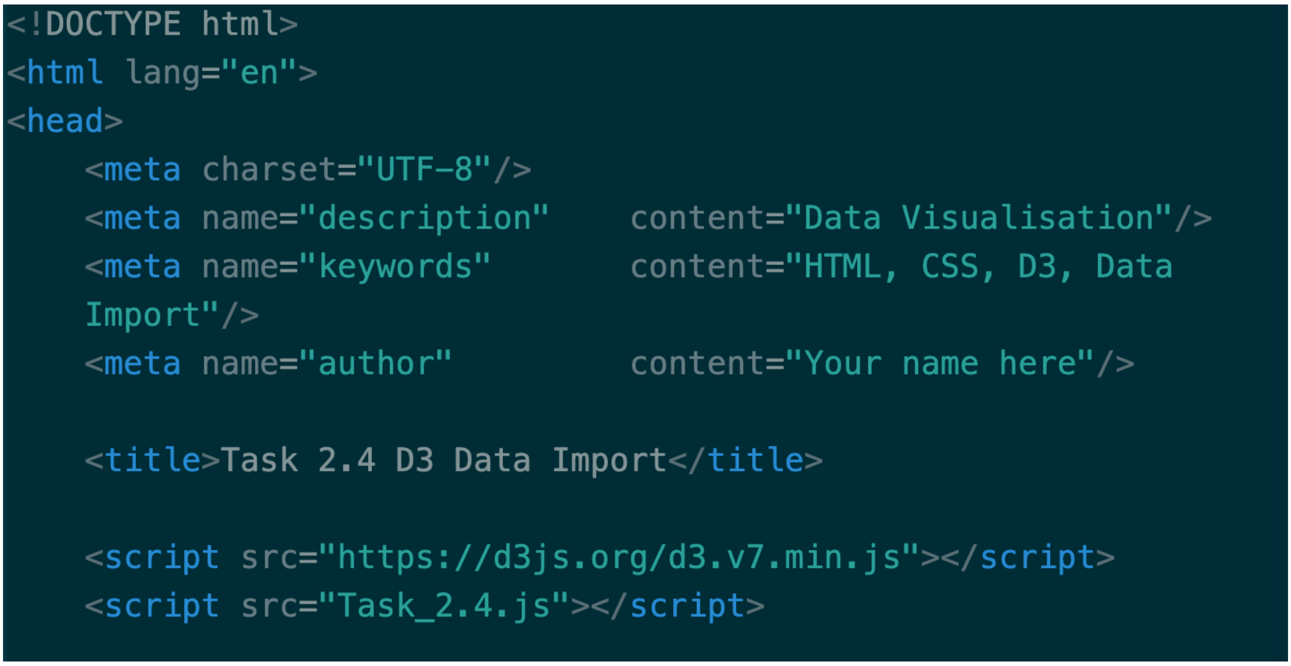
If you haven't done this before, here are some tips….

Create a new file on your text editor and save it as a .js file. Write a function to run when the window loads and add your chart code into it.



```
function init() {

// JavaScript code to go here

}
window.onload = init;
```

Don’t forget to call your new .js file in the header of your HTML file.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <meta name="description"    content="Data Visualisation"/>
    <meta name="keywords"       content="HTML, CSS, D3, Data
    Import"/>
    <meta name="author"         content="Your name here"/>

    <title>Task 2.4 D3 Data Import</title>

    <script src="https://d3js.org/d3.v7.min.js"></script>
    <script src="Task_2.4.js"></script>
```

See optional exercise on next page

Optional

Try plotting the pet ownership data for 2019 and 2021 in two separate charts. Add labels and figure captions to the two charts. Make sure you start a separate set files for this exercise.

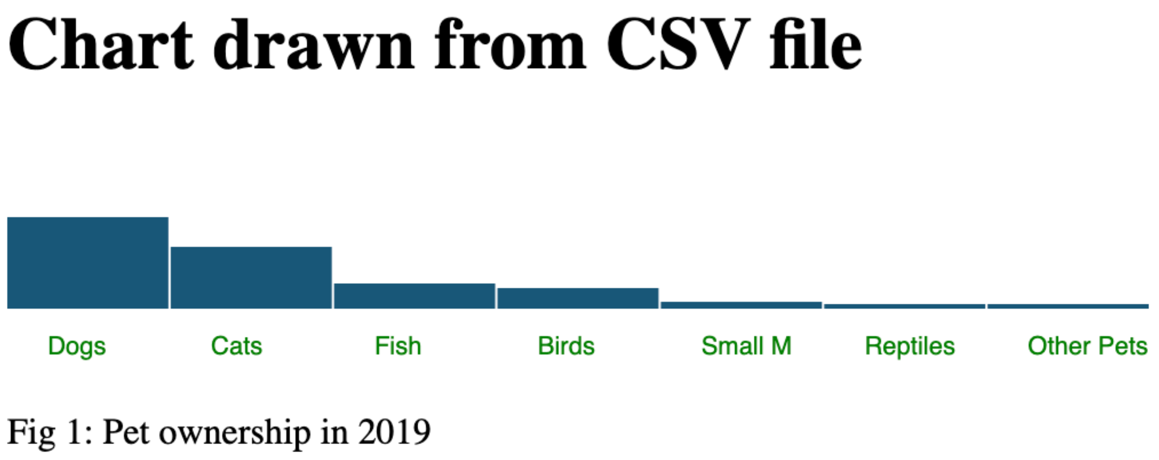


Chart drawn from CSV file

Fig 1: Pet ownership in 2019

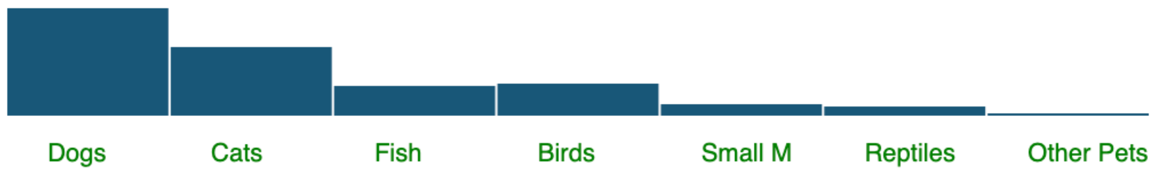
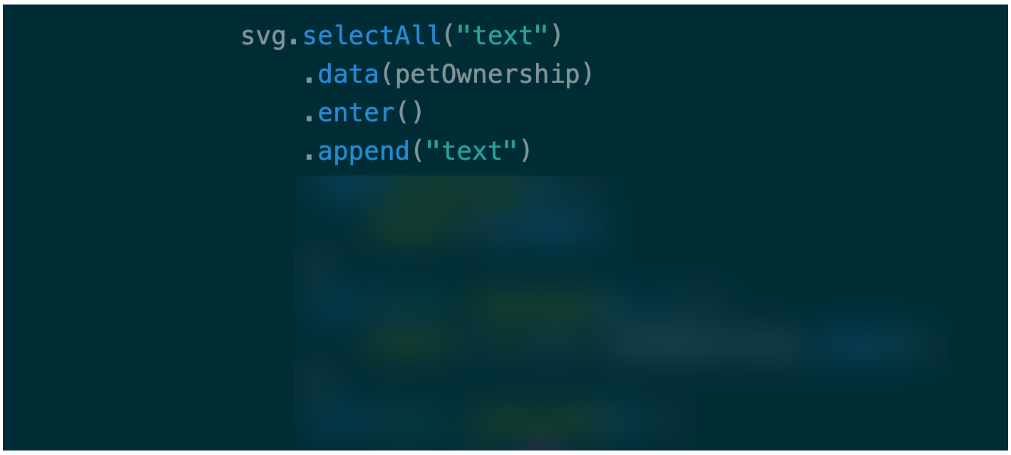


Fig 2: Pet ownership in 2021

A snippet of code to get you started on adding labels…



```
svg.selectAll("text")
    .data(petOwnership)
    .enter()
    .append("text")
```