



# COS30045 Data Visualisation

## Task 2.1 D3 Binding and Drawing with Data

<b>ILO</b>	Create web-based interactive visualisations using real-world data sets.
<b>Aim:</b>	Use D3 to generate elements on a webpage
<b>Resources:</b>	<i>Textbook:</i> Murray Ch 5 <a href="#">Murray on ProQuest</a> <a href="#">Murray on Safari</a>
<b>To be marked as Complete your submission must:</b>	Submit working code that meets the requirements specified in document below. Demonstrate appropriate use of HTML, CSS and D3. Properly formatted code Well commented code with references to code sourced from web, stack overflow etc. where appropriate. Demonstrate and explain code to tutor in class.
<b>Submission</b>	Submit to Doubtfire <ul style="list-style-type: none"><li>• screenshot of final webpage and annotated DOM</li><li>• code</li></ul> Bring code to class to demonstrate to tutor

## Overview

In this tutorial we will start using D3.

**Note:** This Task Guide is not meant to be fully explanatory. You will also need to work through the examples in the text book *Interactive Data Visualisation for the Web* by Murray.

## Step 1: Start a basic HTML template with D3

Firstly you need to set up a basic HTML template and add in reference to the D3 library in the header. You access the D3 library by storing a copy of the D3 library in your program folder or by using the web link. The second method will make sure you are always using the most up to date version of D3.

```

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

  <title>Task 2.1 D3 Data Binding</title>

  <script src="https://d3js.org/d3.v5.min.js"></script>

</head>
<body>

  <h1>The D3 Journey starts here...</h1>

  <script>

    //D3 Code goes here

  </script>

```

reference to v5 of D3 library

## Step 2 Generating new page elements with D3

Replace the `//D3 code goes here` with the following D3 code and run the web page.

```

<body>

  <h1>The D3 Journey starts here...</h1>

  <script>
    //example from Murray

    d3.select("body")
      .append("p")
      .text("New paragraph!");

  </script>

  <br>
  <bf>
  <footer style="color:grey">COS30045 Data Visualisation<br>
    Joe Bloggs</footer>
</body>

```

D3 methods

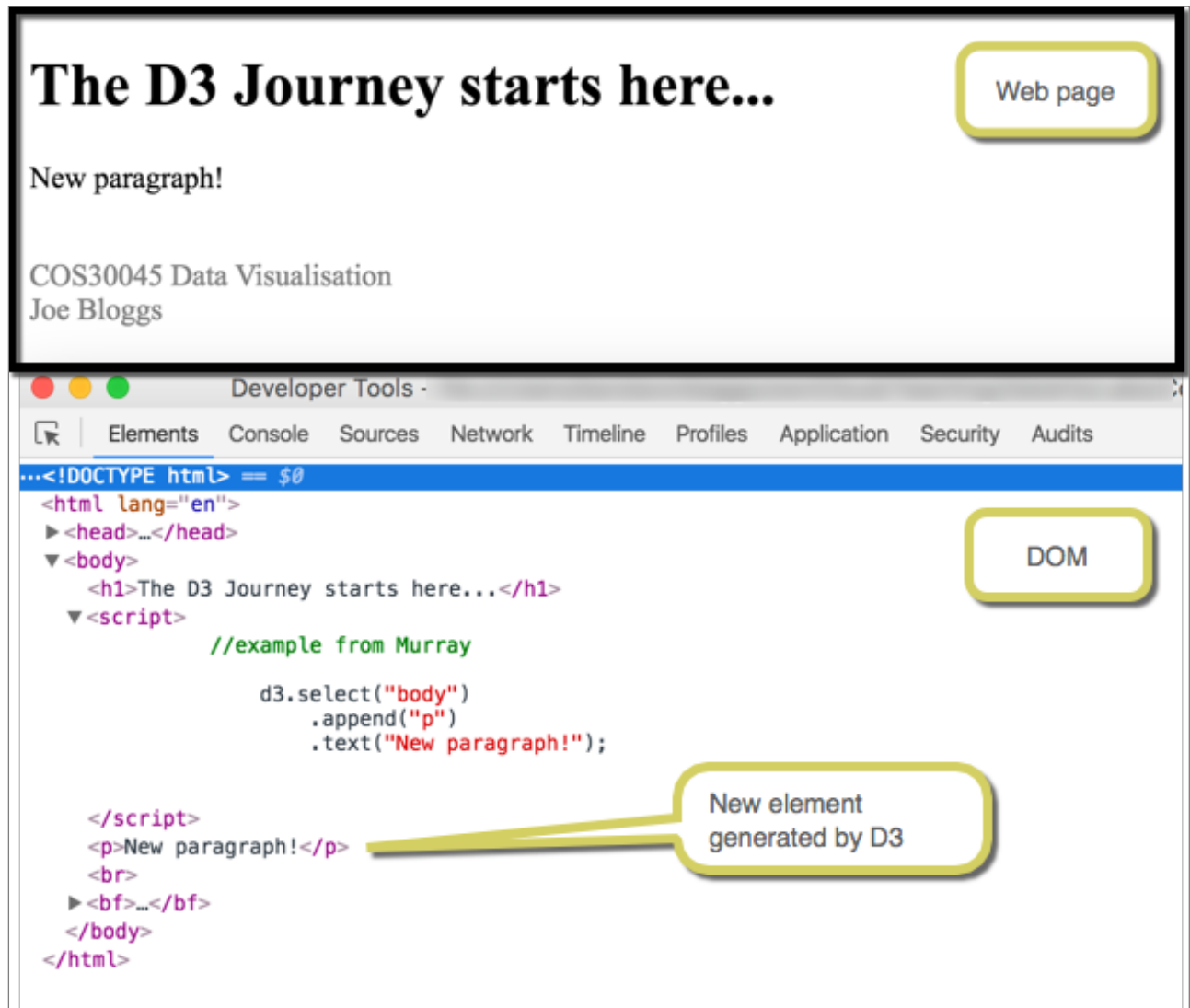
Select the body of the document.

Append a paragraph element to the element selected previously (i.e., the body).

Insert the text into current selection (i.e., the newly generated `<p>` `</p>` tags).

Finish command with ;

If you inspect the DOM you will see that D3 has added (i.e., appended) an a new p element and filled it with the specified text.



### Step 3 Binding data to page elements

D3 allows us to generate page elements and map them to data. We can then customise some aspect of the page element (e.g., a rectangle, circle etc) to reflect some aspect of the data (e.g., height of rectangle, area of circle, colour, thickness of line etc).

Just after the script tag, create a data variable containing numbers between 1 and 30. We will now bind the data set to a set of corresponding p elements.

The screenshot shows a code editor with the following JavaScript code:

```
<script>
//example from Murray

var dataset = [14, 5, 26, 23, 9];

d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text("New Paragraph");
</script>
```

Callouts explain the code:

- Selects all p (although they don't yet exist)**: Points to `d3.select("body").selectAll("p")`.
- Counts and prepares the data values**: Points to `.data(dataset)`.
- Creates a new placeholder element for each bit of data**: Points to `.enter()`.
- Appends a p element to match each placeholder**: Points to `.append("p")`.
- Creates a new placeholder element for each bit of data**: Points to `.text("New Paragraph");`.

Below the code, the browser output shows five "New Paragraph" elements. A callout **Takes the new p elements and inserts a text value** points to the first element. Another callout **5 p elements to match our 5 data points** points to the list of elements.

COS30045 Data Visualisation  
Joe Bloggs

At the moment the data is attached to the p (you can see this by checking the DOM outputting the elements to the in the console using:

```
console.log(d3.selectAll("p"));
```

However, the data is currently not being visualised in the display. Let's get the data to print to the web page:

The screenshot shows the same code as before, but with an anonymous function for the `.text()` method:

```
<script>

var dataset = [14, 5, 26, 23, 9];
//example from Murray

d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text(function(d) {
    return d;
  });
</script>
```

A callout **Anonymous function - accepts d from the data set as an input and loops through each value** points to the function definition.

Use some selection rules (e.g., if, else, then...etc) and html formatting to produce something like the following:

## Creating and Formatting Paragraph Elements with D3

Warning: Joe watched 14 cat videos today.

Joe watched 5 cat videos today.

Warning: Joe watched 26 cat videos today.

Warning: Joe watched 23 cat videos today.

Joe watched 9 cat videos today.

COS30045 Data Visualisation  
Joe Bloggs

**Tip:** You can use both `d` and the index `i` in your anonymous functions. For example:

```
.text(function(d, i) {  
    return "i = " + i + " d = "+d;  
})
```

Will give you the following out put:

```
i = 0 d = 14  
i = 1 d = 5  
i = 2 d = 26  
i = 3 d = 23  
i = 4 d = 9
```

### Requirements

Your submission must include:

- code demonstrating
  - standard HTML template with appropriate meta data and page title
  - the use of D3 to bind data values from a data set to html elements and display on a webpage
  - the use the data to manipulate some visual property of the html element
- screenshot of webpage

- annotated screenshot of DOM showing D3 generated elements (i.e., show the p elements, the indexes and the data in a p element). To view in the console use:

```
console.log(d3.selectAll("p"));
```