02. D3JS Selections.

Selections is one of the core concepts in D3.js. It is based on CSS selectors. It allows us to select one or more elements in a webpage. In addition, it allows us to modify, append, or remove elements in a relation to the pre-defined dataset. In this chapter, we will see how to use selections to create data visualizations.

D3.js helps to select elements from the HTML page using the following two methods −

- **select**() − Selects only one DOM element by matching the given CSS selector. If there are more than one elements for the given CSS selector, it selects the first one only.
- **selectAll**() − Selects all DOM elements by matching the given CSS selector. If you are familiar with selecting elements with jQuery, D3.js selectors are almost the same.

2.1. The select() method

The **select**() method selects the HTML element based on CSS Selectors. In CSS Selectors, you can define and access HTML-elements in the following three ways:

- **Tag** of a HTML element (e.g. div, h1, p, span, etc.,)
- **Class name** of a HTML element
- **ID** of a HTML element

Let us see it in action with examples.

Selection by Tag

You can select HTML elements using its TAG.

Example: The following syntax is used to select the "div" tag elements,

```
d3.select("div")
```

Example: Create a page "select_by_tag.html" and add the following changes

```
<!DOCTYPE html>
<html>
   <head>
      <script src = "https://d3js.org/d3.v5.min.js"></script>
   </head>
   <body>
      <div>
         Hello World!
      </div>
      <script>
         alert(d3.select("div").text());
      </script>
   </body>
</html>
```

By requesting the webpage through the browser, you will see the following output on the screen: *Hello World!*

Selection by Class name

HTML elements styled using CSS classes can be selected by using the following syntax.

Example: The following syntax is used to select all elements with a class,

```
d3.select(".<class name>")
```

Example: Create a webpage "select_by_class.html" and add the following changes –

```
<!DOCTYPE html>
<html>
   <head>
      <script src = "https://d3js.org/d3.v5.min.js"></script>
   </head>
   <body>
      <div class = "myclass">
         Hello World!
      </div>
      <script>
         alert(d3.select(".myclass").text());
      </script>
   </body>
</html>
```

By requesting the webpage through the browser, you will see the following output on the screen: Hello World!

Selection by ID

Every element in a HTML page should have a unique ID. We can use this unique ID of an element to access it using the **select**() method as specified below.

Example: The following syntax is used to select an element with id,

```
d3.select(".<class name>")d3.select("#<id of an element>")
```

Example: Create a webpage "select_by_id.html" and add the following changes.

```
<!DOCTYPE html>
<html>
   <head>
```

```
        <script src = "https://d3js.org/d3.v5.min.js"></script>
    </head>
    <body>
        <div class = "myclass">
            Hello World!
        </div>
        <script>
            alert(d3.select(".myclass").text());
        </script>
    </body>
</html>
```

By requesting the webpage through the browser, you will see the following output on the screen: *Hello World!*

2.2. Adding DOM Elements

The *D3.js* selection provides the **append**() and the **text**() methods to append new elements into the existing HTML documents. This section explains about adding DOM elements in detail.

The append() Method

The **append**() method appends a new element as the last child of the element in the current selection. This method can also modify the style of the elements, their attributes, properties, HTML and text content.

Example: Create a webpage "select_and_append.html" and add the following changes

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <script src = "https://d3js.org/d3.v5.min.js"></script>
    </head>
    <body>
        <div class = "myclass">
            Hello World!
        </div>
        <script>
            d3.select("div.myclass").append("span");
        </script>
    </body>
</html>
```

Here, the append() method adds a new tag span inside the div tag as shown below.

Example:

```
<div class = "myclass">
    Hello World!<span></span>
</div>
```

The text() Method

The text() method is used to set the content of the selected / appended elements. Let us change the above example and add the text() method as shown below.

Example:

```
<!DOCTYPE html>
<html>
    <head>
        <script src = "https://d3js.org/d3.v4.min.js"></script>
```

```
        </head>
        <body>
           <div class = "myclass">
              Hello World!
           </div>
           <script>
              d3.select("div.myclass").append("span").text("from D3.js");
           </script>
        </body>
     </html>
```

Now refresh the webpage and you will see the following response: *Hello World! From D3.js*

Here, the above script performs a chaining operation. D3.js smartly employs a technique called the chain syntax, which you may recognize from jQuery. By chaining methods together with periods, you can perform several actions in a single line of code. It is fast and easy. The same script can also access without chain syntax as shown below.

Example:

```
var body = d3.select("div.myclass");
var span = body.append("span");
span.text("from D3.js");
```

2.3. Modifying Elements

D3.js provides various methods, ***html(), attr() and style()*** to modify the content and style of the selected elements. Let us see how to use modify methods in this chapter.

The html() Method

The **html**() method is used to set the html content of the selected / appended elements.

Example: Create a webpage "select_and_add_html.html" and add the following code.

```
<!DOCTYPE html>
<html>
   <head>
      <script src = "https://d3js.org/d3.v4.min.js"></script>
   </head>
   <body>
      <div class = "myclass">
         Hello World!
      </div>
      <script>
         d3.select(".myclass").html("Hello World! <span>from
D3.js</span>");
      </script>
   </body>
</html>
```

By requesting the webpage through the browser, you will see the following output on the screen: *Hello World! from D3.js*

The attr() Method

The attr() method is used to add or update the attribute of the selected elements.

Example: Create a webpage "select_and_modify.html" and add the following code.

```
<!DOCTYPE html>
<html>
   <head>
      <script src = "https://d3js.org/d3.v4.min.js"></script>
   </head>
```

```
        <body>
            <div class = "myclass">
                Hello World!
            </div>
            <script>
                d3.select(".myclass").attr("style", "color: red");
            </script>
        </body>
    </html>
```

By requesting the webpage through the browser, you will see the following output on the screen: *Hello World!* in red color.

The style() Method

The style() method is used to set the style property of the selected elements.

Example: Create a webpage "select_and_style.html" and add the following code.

```
<!DOCTYPE html>
<html>
    <head>
        <script src = "https://d3js.org/d3.v4.min.js"></script>
    </head>
    <body>
        <div class = "myclass">
            Hello World!
        </div>
        <script>
            d3.select(".myclass").style("color", "red");
        </script>
    </body>
</html>
```

By requesting the webpage through the browser, you will see the following output on the screen: *Hello World!* in red color.

The classed() Method

The classed() method is exclusively used to set the "class" attribute of an HTML element. Since, a single HTML element can have multiple classes; we need to be careful while assigning a class to an HTML element. This method knows how to handle one or many classes on an element, and it will be performant.

Add class − To add a class, the second parameter of the classed method must be set to true. It is defined below −

Example:

```
d3.select(".myclass").classed("myanotherclass", true);
```

Remove class − To remove a class, the second parameter of the classed method must be set to false. It is defined below −

Example:

```
d3.select(".myclass").classed("myanotherclass", false);
```

Check class − To check for the existence of a class, just leave off the second parameter and pass the class name you are querying. This will return true, if it exists, false, if it does not.

Example:

```
d3.select(".myclass").classed("myanotherclass");
```

This will return true, if any element in the selection has the class. Use d3.select for single element selection.

Toggle class − To flip a class to the opposite state – remove it if it exists already, add it if it does not yet exist – you can do one of the following. For a single element, the code might look as shown below:

Example:

```
var element = d3.select(".myclass")
element.classed("myanotherclass", !oneBar.classed("myanotherclass"));
```

2.4. The selectAll() Method

The **selectAll**() method is used to select multiple elements in the HTML document. The select method selects the first element, but the *selectAll* method selects all the elements that match the specific selector string. In case the selection matches none, then it returns an empty selection. We can chain all the appending modifying methods, **append(), html(), text(), attr(), style(), classed()**, etc., in the **selectAll**() method as well. In this case, the methods will affect all the matching elements.

Example: Create a new webpage "*select_multiple.html*" and add the following script

```
<!DOCTYPE html>
<html>
   <head>
      <script src = "https://d3js.org/d3.v4.min.js"></script>
   </head>
```

```
        <body>
            <h2 class = "myclass">Message</h2>
            <div class = "myclass">
               Hello World!
            </div>
            <script>
               d3.selectAll(".myclass").attr("style", "color: red");
            </script>
        </body>
    </html>
```

By requesting the webpage through the browser, you will see the following output on the screen.

Message

Hello World!

Here, the **attr**() method applies to both **div** and **h2** tag and the color of the text in both tags changes to Red.