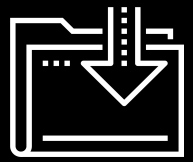
JavaScript with D3.js

Data Boot Camp

Lesson 14.3



Class Objectives

By the end of this lesson, you will be able to:

- Create charts by using data from API calls.
- Use D3 for basic document object model (DOM) manipulation and event handling.
- Apply the `this` keyword to reference elements within a function.
- Dynamically manipulate the DOM through events.
- Manipulate charts through dropdown events and click events.
- Use `Plotly.restyle()` to create dynamic charts.

D3 and D3.json

D3.js is an incredibly powerful visualization library written in JavaScript. Data professionals use it to create interactive data visualizations in browsers.

D3

D3 is a large library with many different subsets, and this class will focus on a subset of D3 for selecting and creating HTML elements dynamically.

Instructor Demonstration

D3.json

Activity: D3.json

In this activity, you will make API calls to SpaceX that return information about the roadster and capsules.

Suggested Time:

15 minutes

Time's Up! Let's Review.

D3 Select and Append

D3 Select & Append

Creates a reference to DOM element with the class `text1`

Captures the `text` of that element

D3 link from d3js.org

index.html

HTML

console

index.js

```
// Select the text of an HTML element
let text1 = d3.select(".text1").text();
console.log("text1 says: ", text1);

let text2 = d3.select("#text2").text();
console.log("text2 says: ", text2);

// Modify the text of an HTML element
d3.select(".text1").text("Hey, I changed this!");

// Capture the HTML of a selection
let myLink = d3.select(".my-link").html();
console.log("my-link: ", myLink);

// Select an element's child element
// An object is returned
let myLinkAnchor = d3.select(".my-link>a");
console.log(myLinkAnchor);

// // Capture the child element's href attribute
let myLinkAnchorAttribute = myLinkAnchor.attr("href");
console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);

// Change an element's attribute
myLinkAnchor.attr("href", "https://python.org");

// Use chaining to join methods
d3.select(".my-link>a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!!");

// Select all list items, then change their font color
d3.selectAll("li").style("color", "blue");

// Create a new element
let li1 = d3.select("ul").append("li");
li1.text("A new item has been added!");

// Use chaining to create a new element and set its text
let li2 = d3.select("ul").append("li").text("Another new item!");
```

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>D3 Select</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>

</head>

<body>
  <h1>This is an H1</h1>
  <div class="text1">This div has a class</div>
  <div id="text2">This div has an id</div>
  <div class="my-link">
    <a href="https://github.com/d3/d3-selection">D3 Home</a>
  </div>
```

```
Elements  Console  Sources  Network  Performance  Memory
top       Filter   Default le
text1 says:  This div has a class
text2 says:  This div has an id
my-link:
    <a href="https://github.com/d3/d3-selection">D3 Home</a>
```

D3 Select & Append

index.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>D3 Select</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>

</head>

<body>
  <h1>This is an H1</h1>
  <div class="text1">This div has a class</div>
  <div id="text2">This div has an id</div>
  <div class="my-link">
    <a href="https://github.com/d3/d3-selection">D3 Home</a>
  </div>

  <div class="deeplink">
    <div class="outer">
      <div class="inner">
        <a href="https://github.com/d3/d3-selection">D3 Select</a>
      </div>
    </div>
  </div>

  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>
</body>
<script src="static/js/index.js"></script>

</html>

```

index.js

```js
// Select the text of an HTML element
let text1 = d3.select(".text1").text();
console.log("text1 says: ", text1);

let text2 = d3.select("#text2").text();
console.log("text2 says: ", text2);

// Modify the text of an HTML element
d3.select(".text1").text("Hey, I changed this!");

// Capture the HTML of a selection
let myLink = d3.select(".my-link").html()
console.log("my-link: ", myLink);

// Select an element's child element
// An object is returned
let myLinkAnchor = d3.select(".my-link>a");
console.log(myLinkAnchor);

// // Capture the child element's href attribute
let myLinkAnchorAttribute = myLinkAnchor.attr("href");
console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);

// Change an element's attribute
myLinkAnchor.attr("href", "https://python.org");

// Use chaining to join methods
d3.select(".my-link>a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!!");

// Select all list items, then change their font color
d3.selectAll("li").style("color", "blue");

// Create a new element
let li1 = d3.select("ul").append("li");
li1.text("A new item has been added!");
```

D3 Select & Append

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>D3 Select</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>

</head>

<body>
  <h1>This is an H1</h1>
  <div class="text1">This div has a class</div>
  <div id="text2">This div has an id</div>
  <div class="my-link">
    <a href="https://github.com/d3/d3-selection">D3 Home</a>
  </div>

  <div class="deeplink">
    <div class="outer">
      <div class="inner">
        <a href="https://github.com/d3/d3-selection">D3 Select</a>
      </div>
    </div>
  </div>

  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>
</body>
<script src="static/js/index.js"></script>

</html>
```

index.html

```js
// Select the text of an HTML element
let text1 = d3.select(".text1").text();
console.log("text1 says: ", text1);

let text2 = d3.select("#text2").text();
console.log("text2 says: ", text2);

// Modify the text of an HTML element
d3.select(".text1").text("Hey, I changed this!");

// Capture the HTML of a selection
let myLink = d3.select(".my-link").html();
console.log("my-link: ", myLink);

// Select an element's child element
// An object is returned
let myLinkAnchor = d3.select(".my-link>a");
console.log(myLinkAnchor);

// // Capture the child element's href attribute
let myLinkAnchorAttribute = myLinkAnchor.attr("href");
console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);

// Change an element's attribute
myLinkAnchor.attr("href", "https://python.org");

// Use chaining to join methods
d3.select(".my-link>a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!!");

// Select all list items, then change their font color
d3.selectAll("li").style("color", "blue");

// Create a new element
let li1 = d3.select("ul").append("li");
li1.text("A new item has been added!");

// Use chaining to create a new element and set its text
let li2 = d3.select("ul").append("li").text("Another new item!");
```

index.js

href *attribute*

Attributes of the element

href *attribute*

href *attribute* value

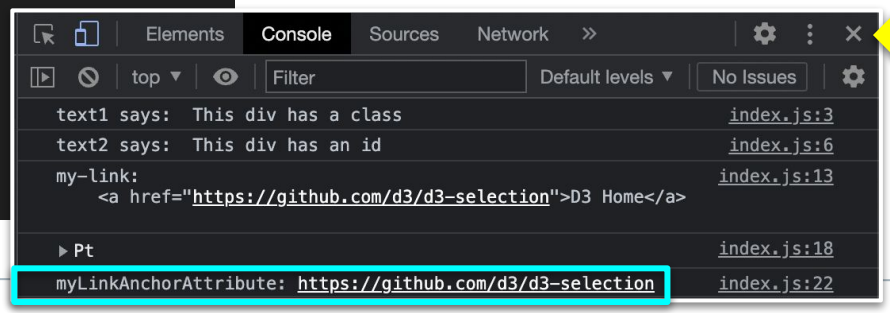
D3 Select & Append

index.js

```
let text1 = d3.select(".text1").text();
console.log("text1 says: ", text1);

let text2 = d3.select("#text2").text();
console.log("text2 says: ", text2);

// Modify the text of an HTML element
d3.select(".text1").text("Hey, I changed this!");

// Capture the HTML of a selection
let myLink = d3.select(".my-link").html();
console.log("my-link: ", myLink);

// Select an element's child element
// An object is returned
let myLinkAnchor = d3.select(".my-link>a");
console.log(myLinkAnchor);

// // Capture the child element's href attribute
let myLinkAnchorAttribute = myLinkAnchor.attr("href");
console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);

// Change an element's attribute
myLinkAnchor.attr("href", "https://python.org");

// Use chaining to join methods
d3.select(".my-link>a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!!");

// Select all list items, then change their font color
d3.selectAll("li").style("color", "blue");

// Create a new element
let li1 = d3.select("ul").append("li");
li1.text("A new item has been added!");

// Use chaining to create a new element and set its text
let li2 = d3.select("ul").append("li").text("Another new item!");
```

The `href` *attribute* of the object

```
// Change an element's attribute
myLinkAnchor.attr("href", "https://python.org");
```



console

Activity: D3 Select

In this activity, you will use D3 to add a new row of data to a table.

Suggested Time:

15 minutes

Time’s Up! Let’s Review.

D3 Table and Event Listeners

What is an event?

An **event** is an action triggered by the user or the browser. JavaScript listens for the event to then execute the code (interact with HTML).

D3 Table and Event Listeners

There are several event types that are supported by the browser, including:

- `click`
- `change`
- `keydown`
- `scroll`
- `pointenter`
- `pointerleave`

```
function handleClick() {
  console.log("A button was clicked!");
  console.log(d3.event.target);
}
```

D3 Table and Event Listeners

Events have two main components:

A target	A reference to the object that dispatched the event.
A handler	A function that executes in response to the event occurring.

```
function handleClick() {
  console.log("A button was clicked!");
  console.log(d3.event.target);
}
```

Instructor Demonstration

Event Listeners

Activity: Button Clicks

In this activity, you will use D3 to create click handlers for upvotes and downvotes.

Suggested Time:

15 minutes

Time’s Up! Let’s Review.


Break

Introducing `this`

In JavaScript, the `this` keyword refers to the object it belongs to. It has different values depending on where it is used.

It can be very resourceful to identify which element triggered an event.

Introducing `this`

Selects all the buttons in the document.

```
d3.selectAll("button").on("click", function() {
 console.log(this);
});
```

Triggers a function that will log `this` to the console.

Introducing `this`

We assign the the `li` element to the variable `listItem` using `d3.select(this)`. Selecting the element with D3 makes it possible to use D3 functions such as style or text on the element.

```
d3.selectAll("li").on("click", function() {
 let listItem = d3.select(this);
 listItem.style("color", "blue");
 let listItemText = listItem.text();
 console.log(listItemText);
});
```

Activity: `this` Button

In this activity, you will refactor the button activity with the `this` keyword.

Suggested Time:

15 minutes

Time’s Up! Let’s Review.

Instructor Demonstration

Dropdowns and Plotly

Activity: Government Expenditure

In this activity, you will enhance their event handling chops by creating a dynamic pie chart using Plotly. When users select a country from the dropdown menu, that country's dataset will display in the browser.

Suggested Time:

15 minutes

Time’s Up! Let’s Review.

Questions?