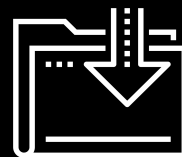




JavaScript with D3.js

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.

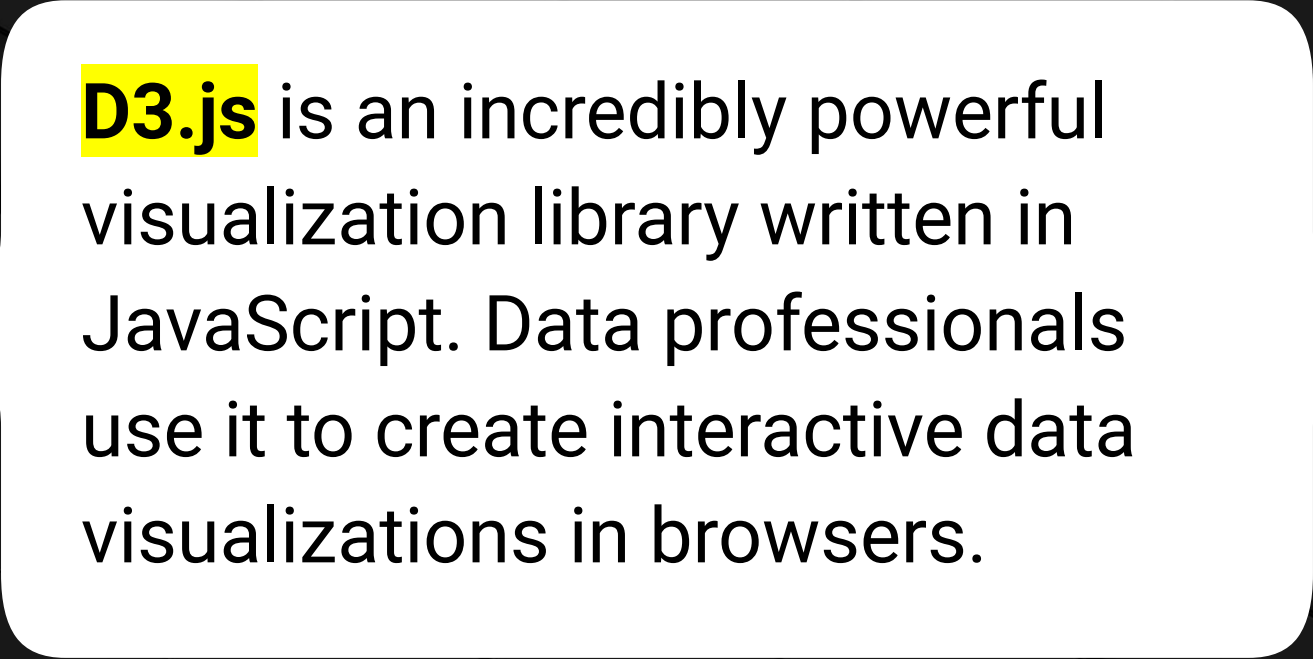


Manipulate charts through dropdown events and click events.



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**

D3 link from d3js.org

Captures the **text** of that element

```
1 // Select the text of an HTML element
2 let text1 = d3.select("#text1").text();
3 console.log("text1 says: ", text1);
4
5 let text2 = d3.select("#text2").text();
6 console.log("text2 says: ", text2);
7
8 // Modify the text of an HTML element
9 d3.select("#text1").text("Hey, I changed this!");
10
11 // Capture the HTML of a selection
12 let myLink = d3.select(".my-link").html();
13 console.log("my-link: ", myLink);
14
15 // Select an element's child element
16 // An object is returned
17 let myLinkAnchor = d3.select(".my-link>a");
18 console.log(myLinkAnchor);
19
20 // Capture the child element's href attribute
21 let myLinkAnchorAttribute = myLinkAnchor.attr("href");
22 console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);
23
24 // Change an element's attribute
25 myLinkAnchor.attr("href", "https://python.org");
26
27 // Use chaining to join methods
28 d3.select(".my-link>a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!!");
29
30 // Select all list items, then change their font color
31 d3.selectAll("li").style("color", "blue");
32
33 // Create a new element
34 let li1 = d3.select("ul").append("li");
35 li1.text("A new item has been added!");
36
37 // Use chaining to create a new element and set its text
38 let li2 = d3.select("ul").append("li").text("Another new item!");
39
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>D3 Select</title>
9   <script src="https://d3js.org/d3.v5.min.js"></script>
10
11 </head>
12
13 <body>
14   <h1>This is an HTML file</h1>
15   <div class="text1">This div has a class</div>
16   <div id="text2">This div has an id</div>
17   <div class="my-link">
18     <a href="https://github.com/d3/d3-selection">D3 Home</a>
19   </div>
20
```

index.html

HTML

Elements Console Sources Network Performance Memory

top Filter Default

text1 says: This div has a class
text2 says: This div has an id

my-link:
D3 Home

console

index.js

D3 Select & Append

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>D3 Select</title>
9   <script src="https://d3js.org/d3.v5.min.js"></script>
10
11 </head>
12
13 <body>
14   <h1>This is an H1</h1>
15   <div class="text1">This div has a class</div>
16   <div id="text2">This div has an id</div>
17   <div class="my-link">
18     <a href="https://github.com/d3/d3-selection">D3 Home</a>
19   </div>
20
21   <div class="deeplink">
22     <div class="outer">
23       <div class="inner">
24         <a href="https://github.com/d3/d3-selection">D3 Select</a>
25       </div>
26     </div>
27   </div>
28
29   <ul>
30     <li>Item 1</li>
31     <li>Item 2</li>
32     <li>Item 3</li>
33   </ul>
34 </body>
35 <script src="static/js/index.js"></script>
36
37 </html>
38
```

index.js

```
1 // Select the text of an HTML element
2 let text1 = d3.select("#text1").text();
3 console.log("text1 says: ", text1);
4
5 let text2 = d3.select("#text2").text();
6 console.log("text2 says: ", text2);
7
8 // Modify the text of an HTML element
9 d3.select("#text1").text("Hey, I changed this!");
10
11 // Capture the HTML of a selection
12 let myLink = d3.select(".my-link").html();
13 console.log("my-link: ", myLink);
14
15 // Select an element's child element
16 // An object is returned
17 let myLinkAnchor = d3.select(".my-link a");
18 console.log(myLinkAnchor);
19
20 // // Capture the child element's href attribute
21 let myLinkAnchorAttribute = myLinkAnchor.attr("href");
22 console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);
23
24 // Change an element's attribute
25 myLinkAnchor.attr("href", "https://python.org");
26
27 // Use chaining to join methods
28 d3.select(".my-link a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!!");
29
30 // Select all list items, then change their font color
31 d3.selectAll("li").style("color", "blue");
32
33 // Create a new element
34 let li1 = d3.select("ul").append("li");
35 li1.text("A new item has been added!");
36
```

D3 Select & Append

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>D3 Select</title>
9   <script src="https://d3js.org/d3.v5.min.js"></script>
10
11 </head>
12
13 <body>
14   <h1>This is an H1</h1>
15   <div class="text1">This div has a class</div>
16   <div id="text2">This div has an id</div>
17   <div class="my-link">
18     <a href="https://github.com/d3/d3-selection">D3 Home</a>
19   </div>
20
21   <div class="deepLink">
22     <div class="outer">
23       <div class="inner">
24         <a href="https://github.com/d3/d3-selection">D3 Select</a>
25       </div>
26     </div>
27   </div>
28
29   <ul>
30     <li>Item 1</li>
31     <li>Item 2</li>
32     <li>Item 3</li>
33   </ul>
34 </body>
35 <script src="static/js/index.js"></script>
36
37 </html>
38
```

index.js

index.html

```
1 // Select the text of an HTML element
2 let text1 = d3.select("text1").text();
3 console.log("text1 says: ", text1);
4
5 let text2 = d3.select("#text2").text();
6 console.log("text2 says: ", text2);
7
8 // Modify the text of an HTML element
9 d3.select("text1").text("Hey, I changed this!");
10
11 // Capture the HTML of a selection
12 let myLink = d3.select(".my-link").html();
13 console.log("my-link: ", myLink);
14
15 // Select an element's child element
16 // An object is returned
17 let myLinkAnchor = d3.select("my-link-a");
18 console.log(myLinkAnchor);
19
20 // Capture the child element's href attribute
21 let myLinkAnchorAttribute = myLinkAnchor.attr("href");
22 console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);
23
24 // Change an element's attribute
25 myLinkAnchor.attr("href", "https://pythons.org");
26
27 // Use chaining to join methods
28 d3.select(".my-link-a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!");
29
30 // Select all list items, then change their font color
31 d3.selectAll("li").style("color", "blue");
32
33 // Create a new element
34 let li1 = d3.select("ul").append("li");
35 li1.text("A new item has been added!");
36
37 // Use chaining to create a new element and set its text
38 let li2 = d3.select("ul").append("li").text("Another new item!");
39
```

href attribute

Pt

console

Attributes
of the
element

href attribute

href attribute value

```
Array(1)
Array(1)
0: a
  __proto__: Array(1)
accessKey: ""
ariaAtomic: null
ariaAutoComplete: null
ariaBusy: null
ariaChecked: null
ariaColCount: null
ariaColIndex: null
ariaColSpan: null
ariaCurrent: null
ariaDescription: null
ariaDisabled: null
ariaExpanded: null
ariaHasPopup: null
ariaHidden: null
ariaKeyShortcuts: null
ariaLabel: null
ariaLevel: null
ariaLive: null
ariaModal: null
ariaMultiline: null
ariaMultiSelectable: null
ariaOrientation: null
ariaPlaceholder: null
ariaPosInSet: null
ariaPressed: null
ariaReadOnly: null
ariaRelevant: null
ariaRequired: null
ariaRoleDescription: null
ariaRowCount: null
ariaRowIndex: null
ariaRowSpan: null
ariaSelected: null
ariaSetSize: null
ariaSort: null
ariaValueMax: null
ariaValueMin: null
ariaValueNow: null
ariaValueText: null
__proto__: Object
attributes: NamedNodeMap
href: href
  __proto__: HTMLAnchorElement
childNodes: NodeList (1)
  firstChild: null
  isConnected: false
  lastChild: null
  localName: "a"
  ownerDocument: https://nytimes.org
  namespaceURI: null
  nextSibling: null
  nodeName: "a"
  nodeType: 1
  nodeValue: "https://nytimes.org"
  ownerDocument: document
  ownerElement: a
  parentElement: null
  parentNode: null
  prefix: null
  previousSibling: null
  content: "https://nytimes.org"
  value: "https://nytimes.org"
```

D3 Select & Append

```
2 let text1 = d3.select(".text1").text();
3 console.log("text1 says: ", text1);
4
5 let text2 = d3.select("#text2").text();
6 console.log("text2 says: ", text2);
7
8 // Modify the text of an HTML element
9 d3.select(".text1").text("Hey, I changed this!");
10
11 // Capture the HTML of a selection
12 let myLink = d3.select("my-link").html();
13 console.log("my-link: ", myLink);
14
15 // Select an element's child element
16 // An object is returned
17 let myLinkAnchor = d3.select("my-link>a");
18 console.log(myLinkAnchor);
19
20 // // Capture the child element's href attribute
21 let myLinkAnchorAttribute = myLinkAnchor.attr("href");
22 console.log("myLinkAnchorAttribute: " + myLinkAnchorAttribute);
23
24 // Change an element's attribute
25 myLinkAnchor.attr("href", "https://python.org");
26
27 // Use chaining to join methods
28 d3.select("my-link>a").attr("href", "https://nytimes.com").text("Now this is a link to the NYT!!");
29
30 // Select all list items, then change their font color
31 d3.selectAll("li").style("color", "blue");
32
33 // Create a new element
34 let li1 = d3.select("ul").append("li");
35 li1.text("A new item has been added!");
36
37 // Use chaining to create a new element and set its text
38 let li2 = d3.select("ul").append("li").text("Another new item!");
```

index.js

The href *attribute* of the object

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

console

Elements	Console	Sources	Network	>>
Filter				
Default levels ▾ No Issues ⚙				
text1 says: This div has a class	index.js:3			
text2 says: This div has an id	index.js:6			
my-link: D3 Home	index.js:13			
▶ Pt	index.js:18			
myLinkAnchorAttribute: https://github.com/d3/d3-selection	index.js:22			



Activity: D3 Select

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

Suggested Time:

15 minutes

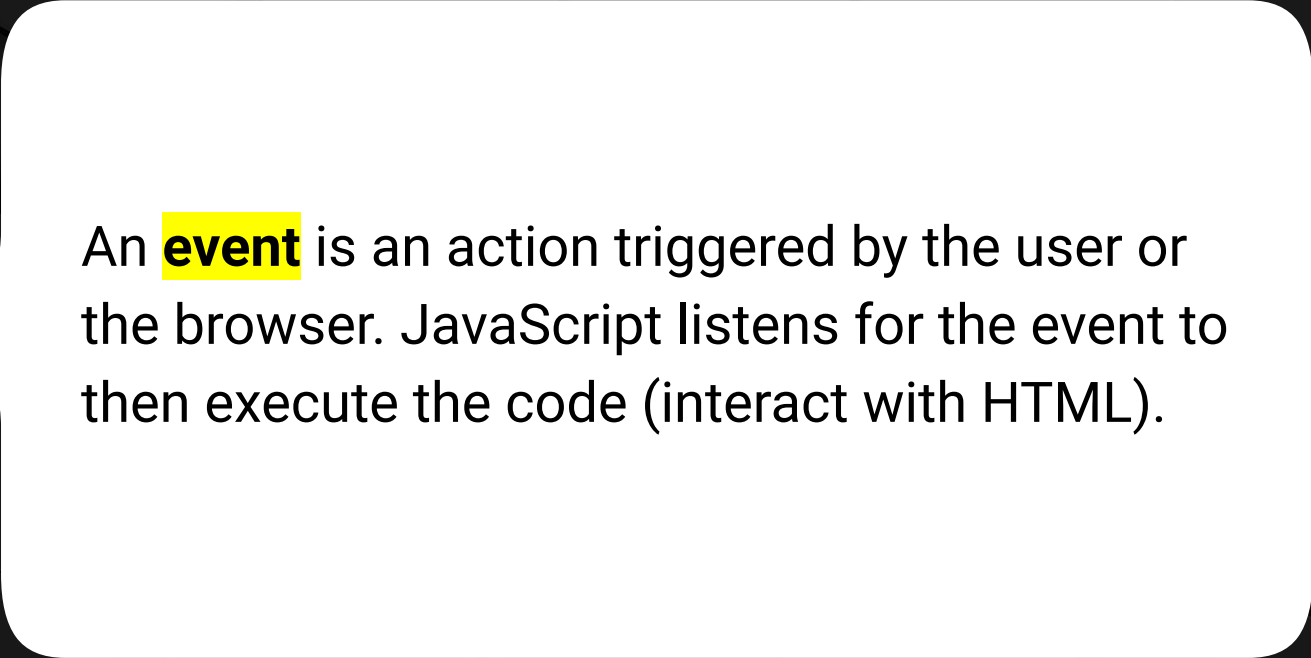


Time's Up! Let's Review.

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



scroll



change



pointerenter



keydown



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

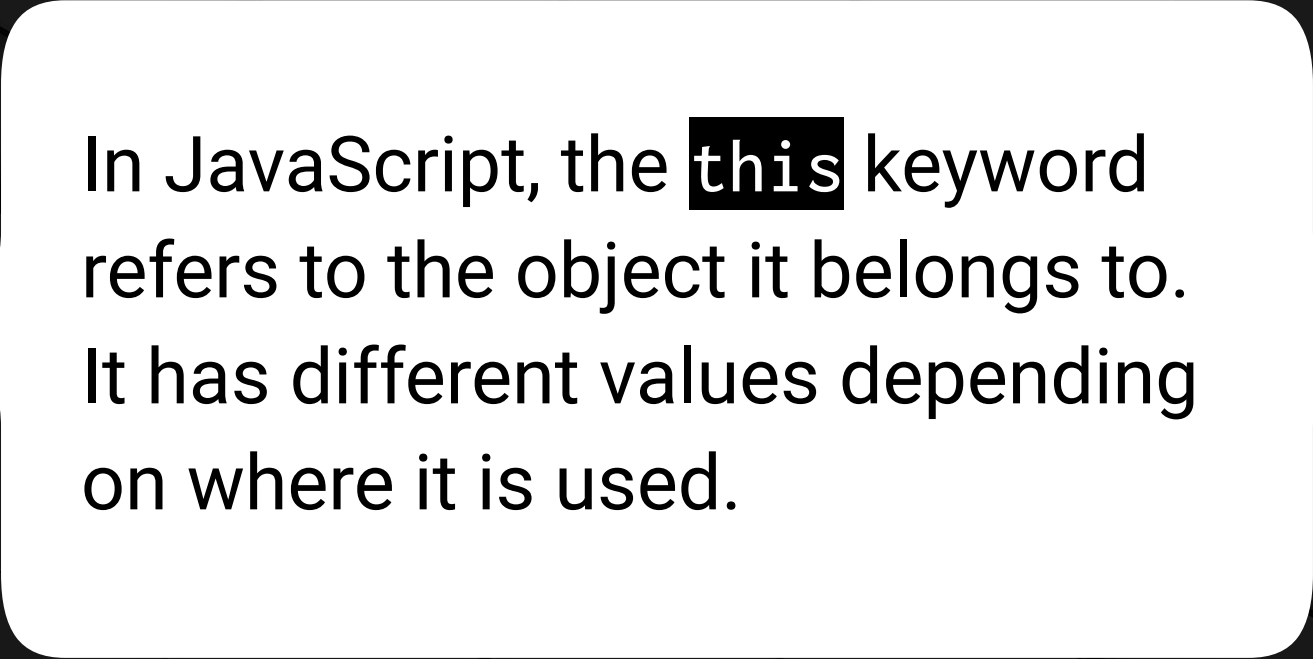


Time's Up! Let's Review.

A close-up, high-angle shot of a computer keyboard. The central focus is a large, white, rectangular key with rounded corners. On this key, there is a dark blue icon of a coffee cup with three wavy lines above it representing steam. Below the icon, the word "Break" is printed in a dark blue, serif font. The key is set against a light-colored, textured keyboard surface. Other keys are visible in the background, including one with a double quote symbol and another with a dash/slash symbol, but they are out of focus.

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?

