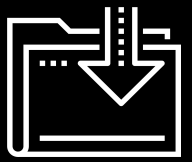
JavaScript with D3.js

Data Boot Camp

Lesson 14.3



Class Objectives

By the end of today's class you will be able to:

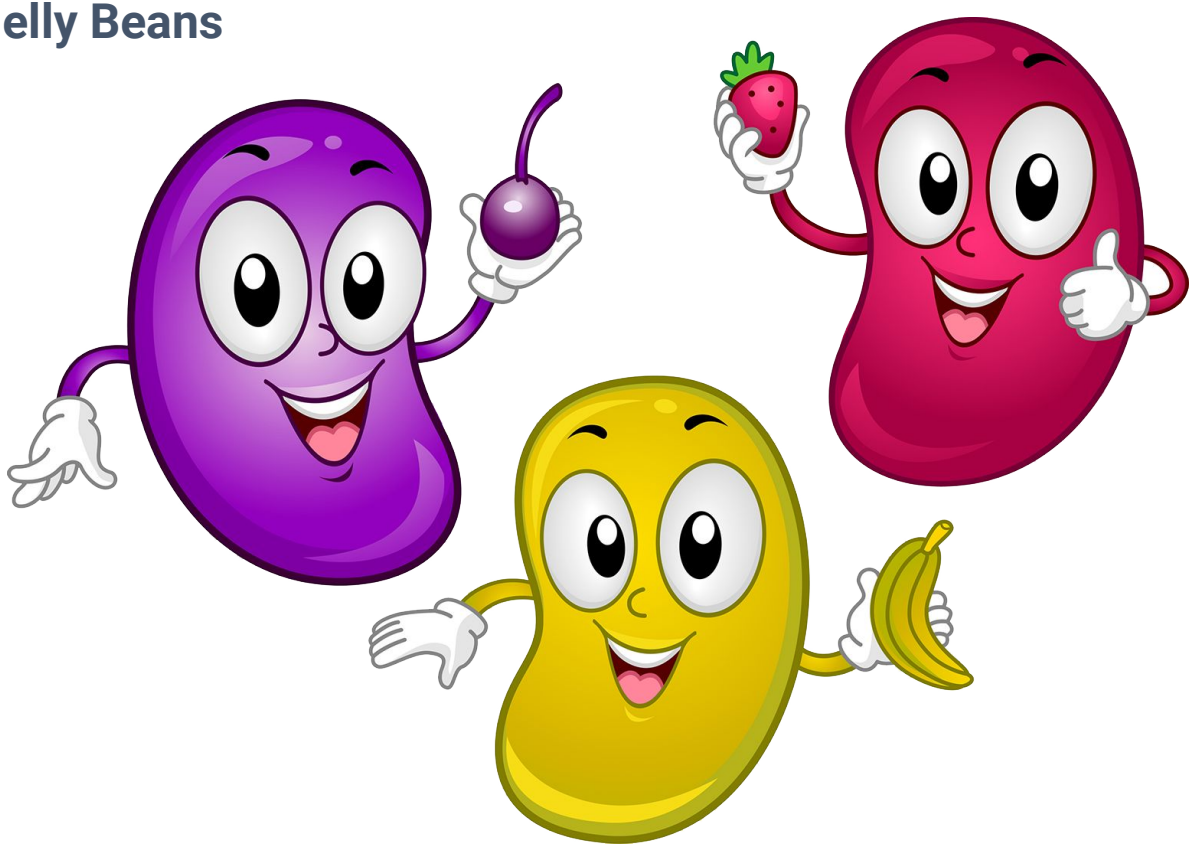
- Use D3 for basic DOM manipulation.
- Populate a table using static data structure.
- Understand events.
- Use `this` to reference elements.
- Use D3 to attach events to DOM elements.
- Dynamically manipulate the DOM through events.
- Dynamically filter tables.

This Will Soon Be You

JavaScript Juggernauts

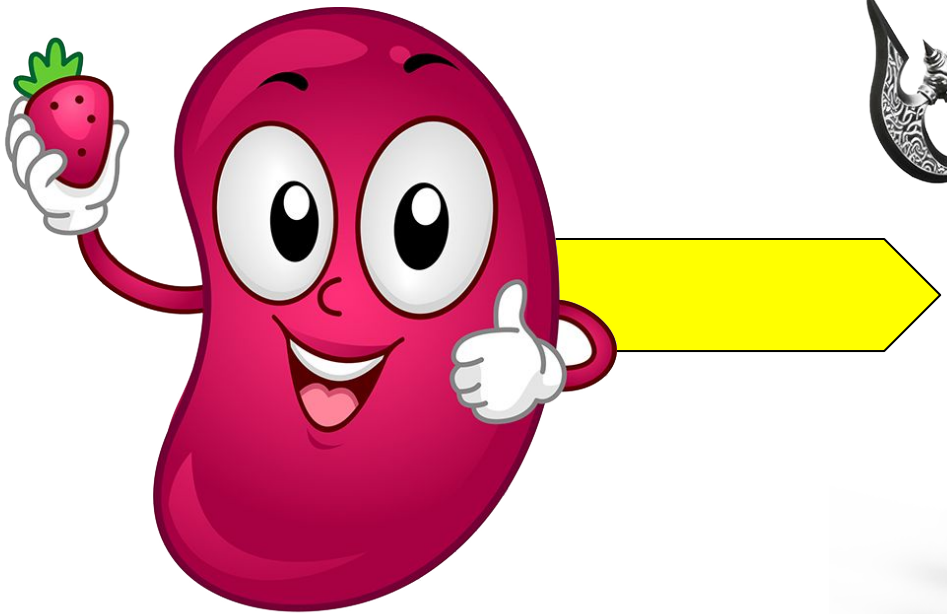
But Right Now You Feel Like:

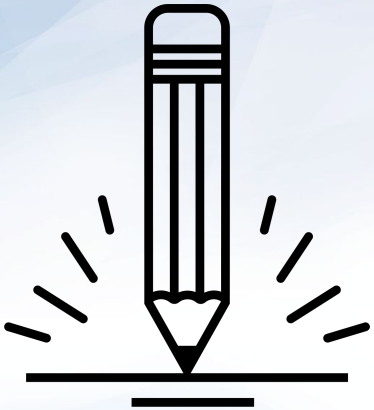
JavaScript Jelly Beans



Transformation to Come!

Hang in there!





Everyone Do: Intro to D3 and Select

In this activity, everyone will be introduced and kick off with **D3** basics.

Suggested Time:
25 Minutes

Everyone Do: Intro to D3 and Select

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>
```

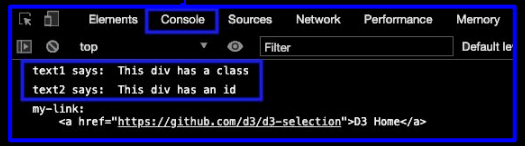
CDN link

index.js*

```js
// Select the text of an HTML element

var text1 = d3.select(".text1").text();

console.log("text1 says: ", text1);

var 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
var myLink = d3.select(".my-link").html();
console.log("my-link: ", myLink);

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

// // Capture the child element's href attribute
var 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.org").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
var 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
var li2 = d3.select("ul").append("li").text("Another new item!");
```

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

Captures the text of that element



Everyone Do: Intro to D3 and Select

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

var text1 = d3.select(".text1").text();

console.log("text1 says: ", text1);

var 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
var myLink = d3.select(".my-link").html();
console.log("my-link: ", myLink);

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

// // Capture the child element's href attribute
var 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.org").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
var 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
var li2 = d3.select("ul").append("li").text("Another new item!");
```

Everyone Do: Intro to D3 and Select

index.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*

```
// Select the text of an HTML element

var text1 = d3.select(".text1").text();

console.log("text1 says: ", text1);

var 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
var myLink = d3.select(".my-link").html();
console.log("my-link: ", myLink);

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

// // Capture the child element's href attribute
var 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.org").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
var 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
var li2 = d3.select("ul").append("li").text("Another new item!");
```

Objected Returned

console

Attributes of the Element

`href` attribute

`href` attribute value

Everyone Do: Intro to D3 and Select

index.js*

```
// Select the text of an HTML element

var text1 = d3.select(".text1").text();

console.log("text1 says: ", text1);

var 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
var myLink = d3.select(".my-link").html();
console.log("my-link: ", myLink);

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

// // Capture the child element's
var myLinkAnchorAttribute = myLi
console.log("myLinkAnchorAttribu

// Change an element's attribute
myLinkAnchor.attr("href", "https://py

// Use chaining to join methods
d3.select(".my-link>a").attr("href", "https://nytimes.org").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
var 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
var li2 = d3.select("ul").append("li").text("Another new item!");
```

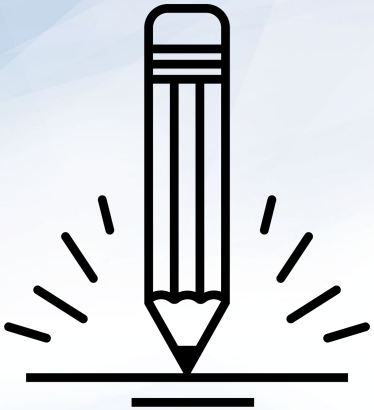
The `href` attribute of the object

tribute
.attr("href");
yLinkAnchorA

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse

top | Filter | Default levels

```
text1 says:  This div has a class                                   index.js:3
text2 says:  This div has an id                                     index.js:6
my-link:                                                            index.js:13
    <a href="https://github.com/d3/d3-selection">D3 Home</a>

▼Pt {_groups: Array(1), _parents: Array(1)}                         index.js:18
  ▶_groups: [Array(1)]
  ▶_parents: [html]
  ▶__proto__: Object
myLinkAnchorAttribute: https://github.com/d3/d3-selection           index.js:22
```

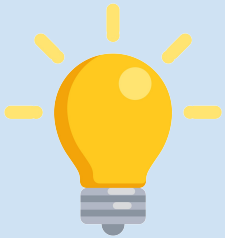


Activity: D3 Select

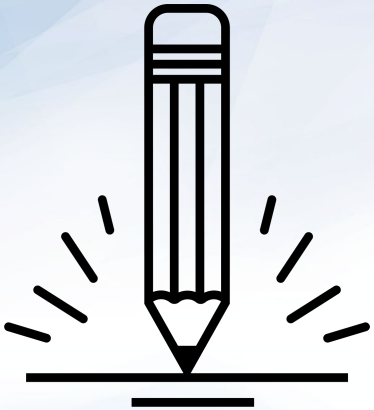
In this activity, you will use **D3** to add a new row of data to a table.

Suggested Time:
20 Minutes

Activity: D3 Select

Instructions	Hints
Use D3 to convert the Bootstrap table into a striped table.	`.table-striped`
Use D3 to select the table body and append a new row and cells for the new student name and grade.	`<tbody>` 

Time’s Up! Let’s Review.

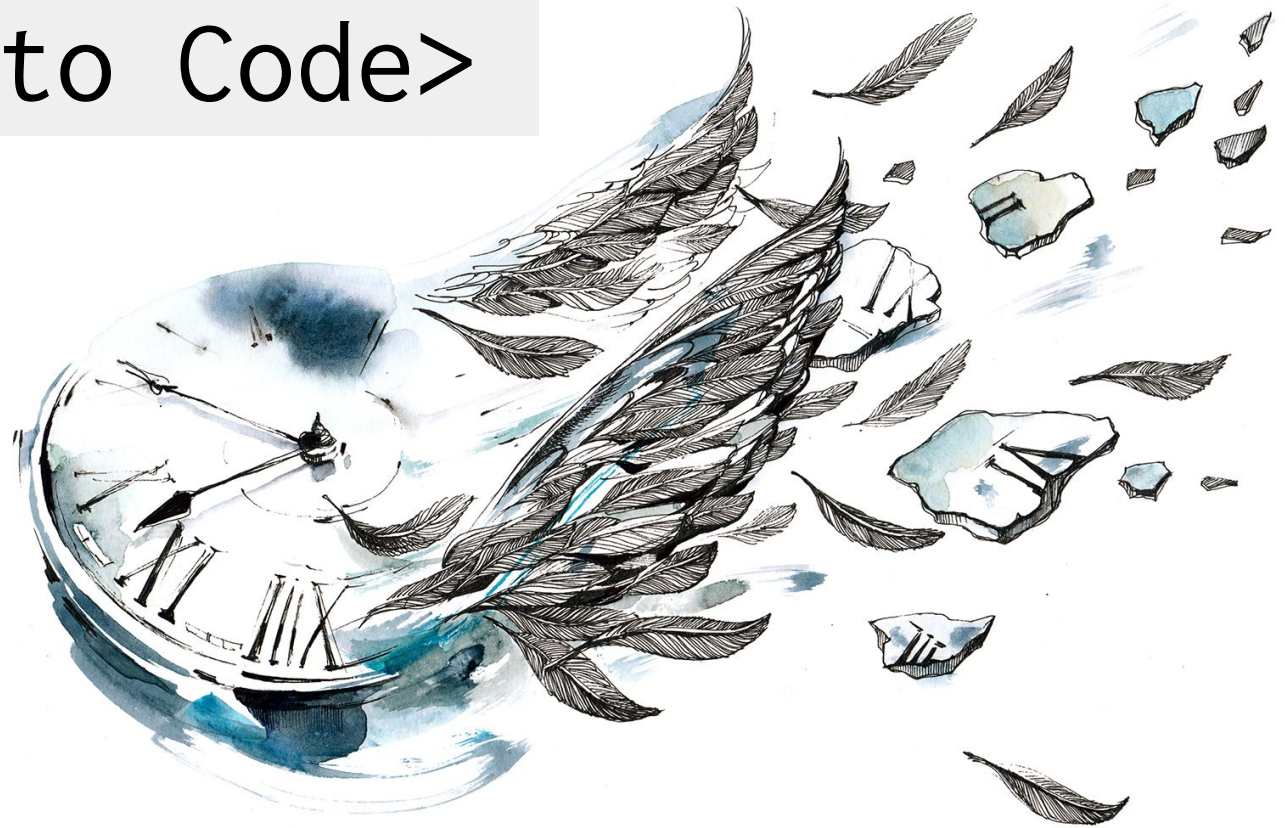


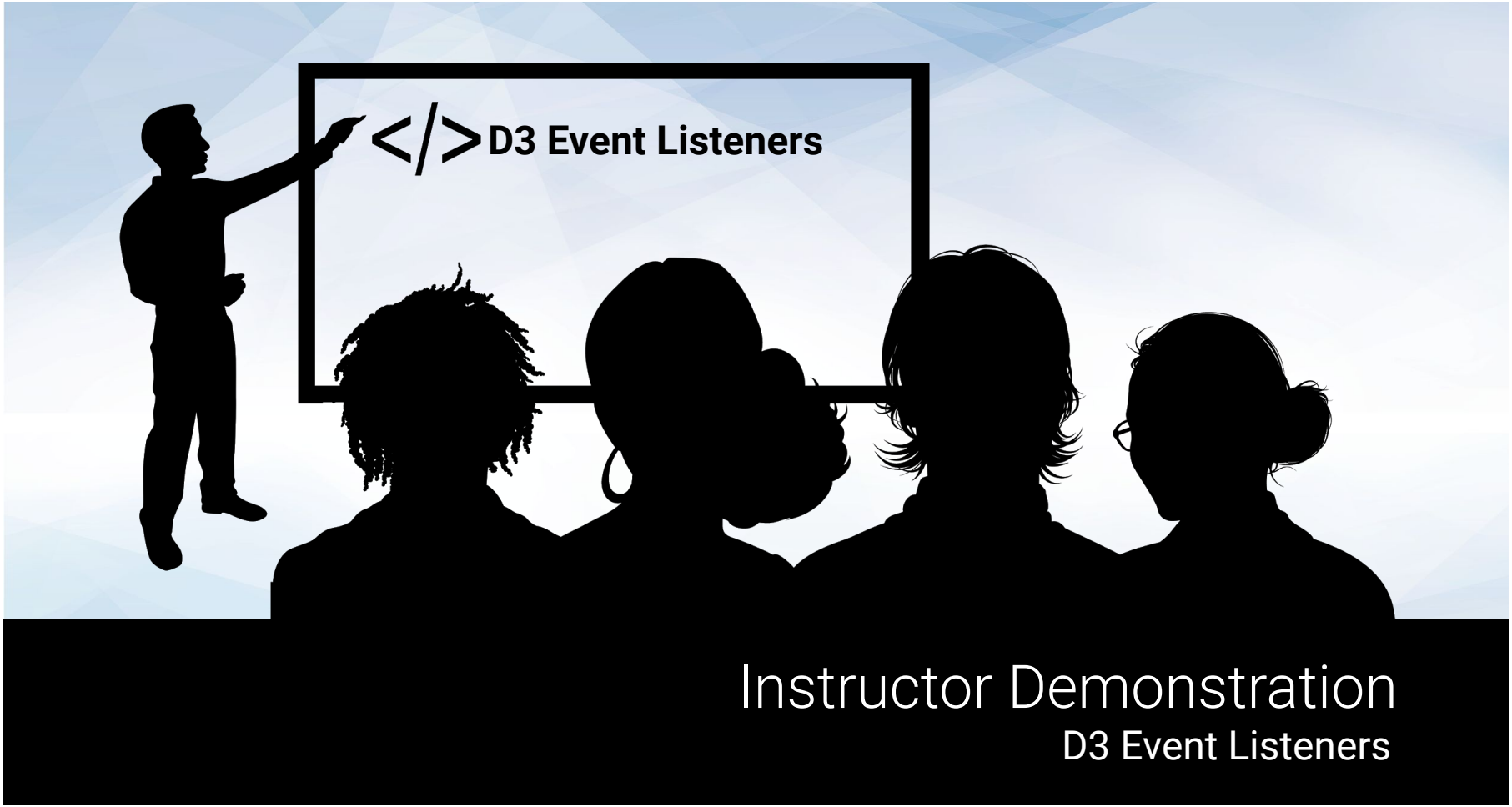
Everyone Do: D3 Table

In this activity, everyone will use D3 to select data and build a raw table.

Suggested Time:
20 Minutes

<Time to Code>





Instructor Demonstration

D3 Event Listeners

What is an Event?

- An action triggered by the user or the browser, detected by JavaScript (listen) to execute the code (interact HTML).
- There are several event types that are supported by the browser, including:
  - `click`
  - `change`
  - `Keydown`
  - `scroll`
  - `pointenter`
  - `pointerleave`

D3 Event Listeners

➔ Events have two main components:

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

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

D3 Event Listeners

- In **D3,** Events are attached using the `.on()` function.

```
button.on("click", handleClick);
```

- Alternatively, the click handler can be defined inline.

```
button.on("click", function() {
 console.log("Hi, a button was clicked!");
 console.log(d3.event.target);
});
```

D3 Event Listeners

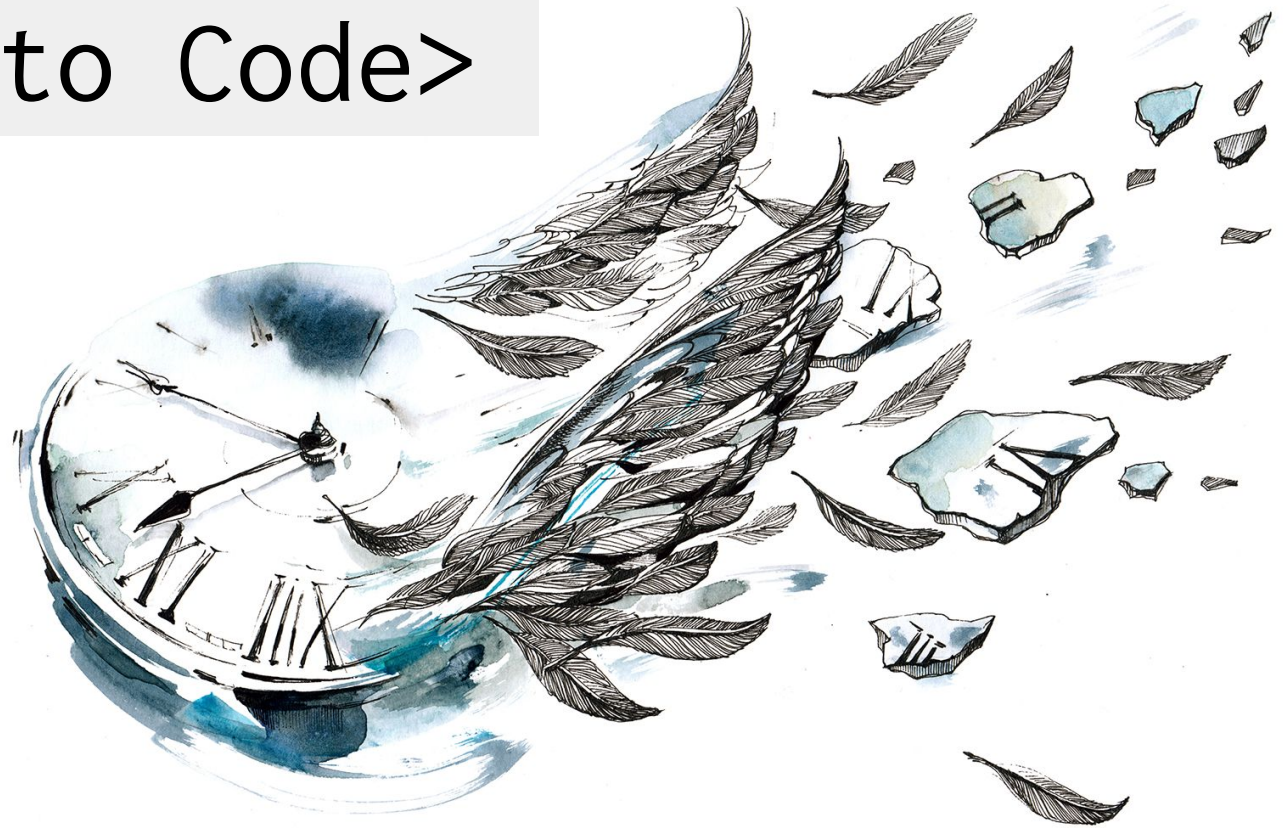
➔ Their are just like functions that can execute code or call other functions.

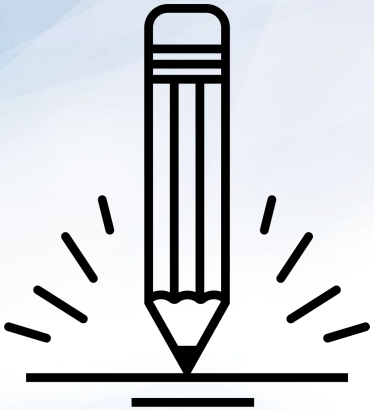
```
button.on("click", function() {
 d3.select(".giphy-me").html("<img src='https://gph.to/2Krfn0w'alt='giphy'>");
});
```

➔ Events change can be tigger by input elements.

```
inputField.on("change", function() {
  var newText = d3.event.target.value;
  console.log(newText);
});
```

<Time to Code>





Activity: On Change

In this activity, you will use **D3** to reverse the input text and display it on the page.

Suggested Time:
20 Minutes

Activity: On Change

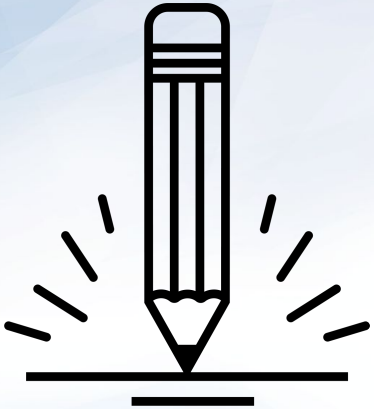
Instructions	Bonus	Hints
Use d3 to select the input (`#text`) and output (`.output`) elements from the page.	Instead of reversing the string, try to calculate the number of characters in the string.	You may need to iterate through the object using `Object.entries` and `forEach`. 
Use d3 to attach an event listener to the input field. This event should call the `handleChange` function any time that the input text changes.		
Finally, complete the `handleChange` function to select the text from the input field and reverse the string. This function will use d3 to set the output element to the value of the reversed string.	Edit the `index.html` file to change the `h1` tag to an unordered list ul. Append each word: `count` as a `li` element.	

Time’s Up! Let’s Review.

Countdown timer

40:00

(with alarm)



Activity: Button Clicks

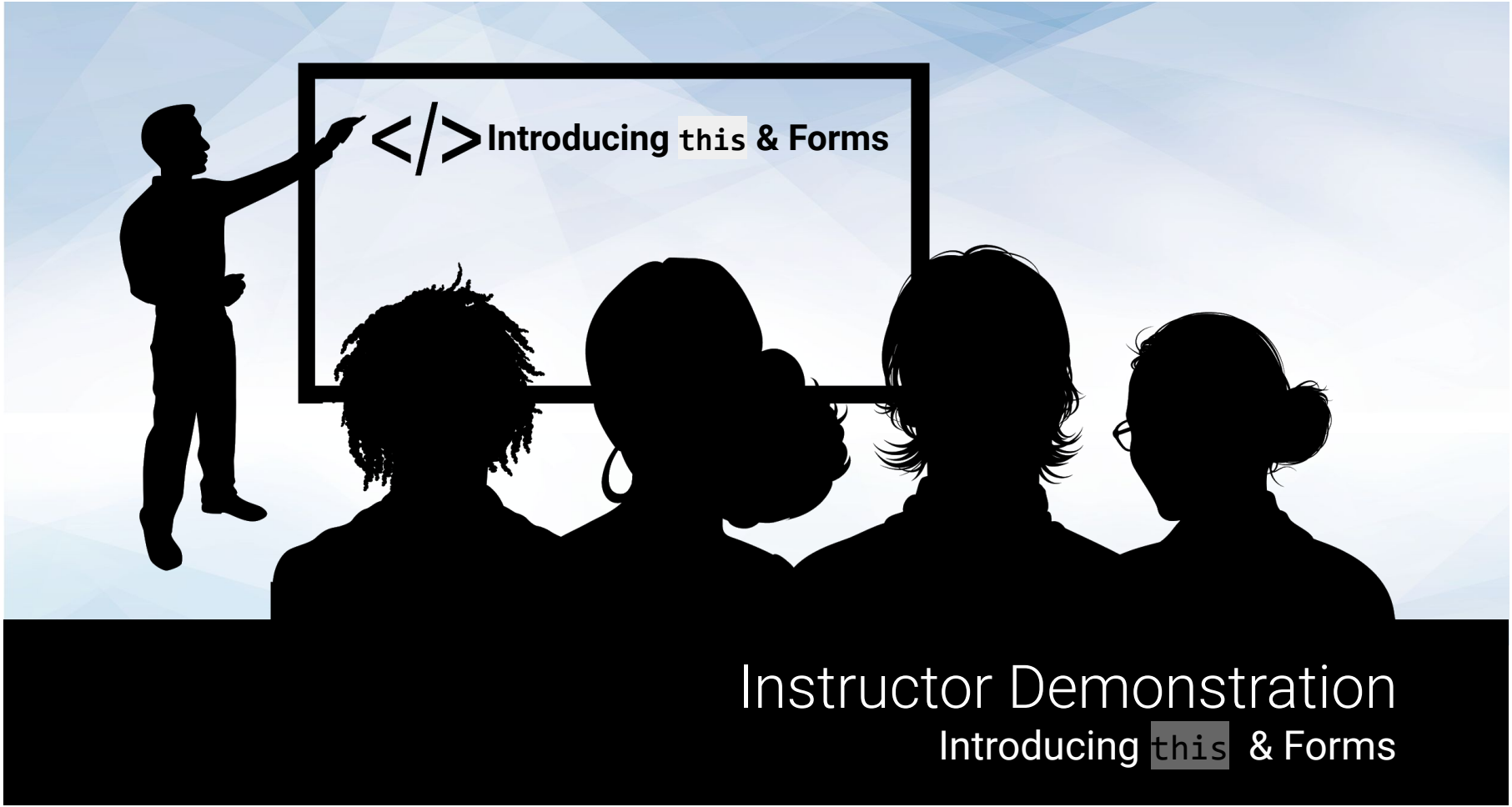
In this activity, you will use **D3** to create click handlers for upvotes and downvotes.

Suggested Time:
20 Minutes

Activity: Button Clicks

Instructions	Bonus	Hints
Use d3 to select the input (`#text`) and output (`.output`) elements from the page.	Instead of reversing the string, try to calculate the number of characters in the string.	You may need to iterate through the object using `Object.entries` and `forEach`. 
Use d3 to attach an event listener to the input field. This event should call the `handleChange` function any time that the input text changes.		
Finally, complete the `handleChange` function to select the text from the input field and reverse the string. This function will use d3 to set the output element to the value of the reversed string.	Edit the `index.html` file to change the `h1` tag to an unordered list ul. Append each word: `count` as a `li` element.	

Time’s Up! Let’s Review.



Instructor Demonstration

Introducing `this` & Forms

Introducing `this` & Form

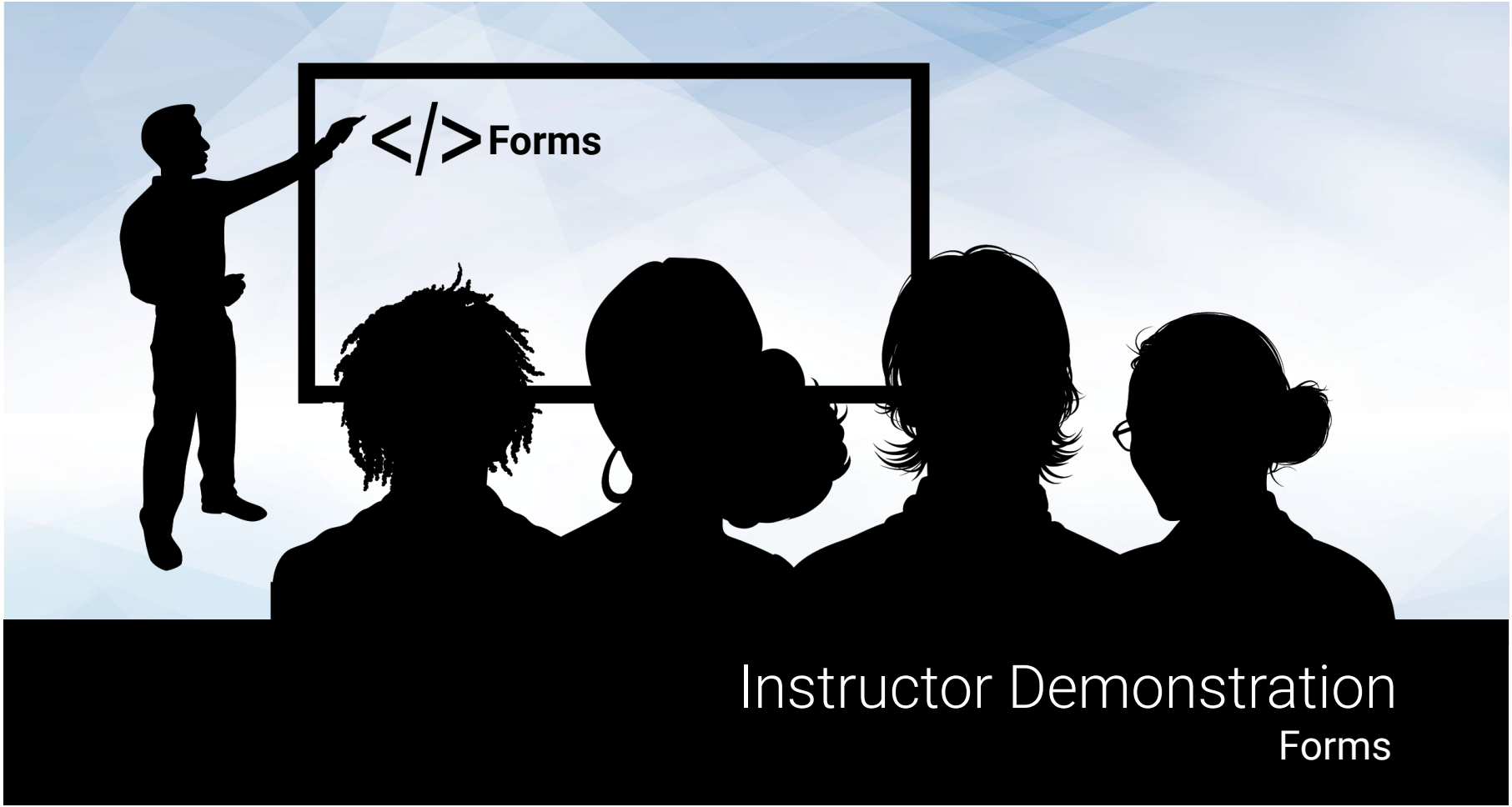
- In JavaScript the `this` keyword refers to the object it belongs to. It has different values depending on where it is used.
- It can be very resourceful to identify which element triggered an event.

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

Selects all the buttons in the document.
A function is triggered that will log `this` to the console.

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

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

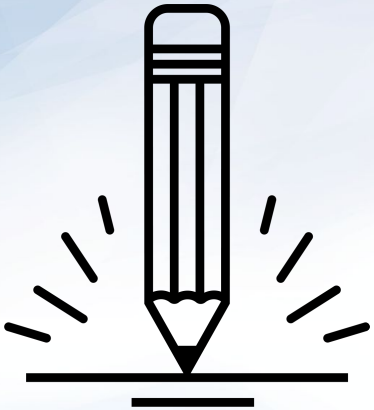


Instructor Demonstration

Forms

Forms

```
var button = d3.select("#button");
var form = d3.select("#form");
button.on("click", runEnter);
form.on("submit",runEnter);

function runEnter() {
 d3.event.preventDefault();
 var inputElement = d3.select("#example-form-input");

 var inputValue = inputElement.property("value");
 console.log(inputValue);
 d3.select("h1>span").text(inputValue);
```



Activity: Form Filter

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

Suggested Time:
25 Minutes

Activity: Form Filter

Instructions	Bonus	Hints 
Use d3 to select `upvote` and `downvote` buttons on the page.	Use an array to save information about each vote: • Store whether it was an "upvote" or "downvote". • Store the current count at each click. • Use an array of arrays or an array of objects to store the data.	Don't forget to use the `.on` function to attach the click handlers to the buttons.
Create click handlers for the upvote and downvote buttons.		You will need one click handler for each button.
The click handlers should do the following: • Select the current vote count from the h3 tag. • Increment or decrement the count depending. on which button was selected. • Update the vote count h3 tag using D3.		You will need to use parseInt to convert the `h3` vote count to a number before you can add or subtract from it.

Time’s Up! Let’s Review.