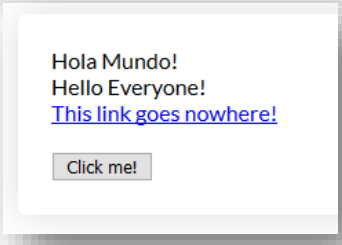
DOM Lab:

- Launch the domLabStartFile.html file
    - The objective of this activity is to change attributes and values on DOM elements
    - Add and remove DOM elements
- The easiest way to select elements is by their id using “getElementById” method on the document object (document.getElementById). This returns a SINGLE element.
    - let container = document.getElementById("container");
- Another method called “querySelector”, selects a SINGLE element using CSS selectors. If multiple elements match the query you pass in to the querySelector, the function will return the first matching element that it finds.
    - let container = document.querySelector("#container");
- To select multiple elements, getElementsByTagName or getElementsByClassName, or querySelectorAll, are functions that can be used.
    - let divs = document.getElementsByTagName("div");
    - let divs = document.querySelectorAll("div");
    - let divsWithClassOfHello = document.getElementsByClassName("hello");
    - let divsWithClassOfHello = document.querySelectorAll(".hello");
- **Modify Properties and Attributes on Elements in the DOM**
    - You can change the text of an element through the “innerHTML” property

```
<script>
    // you can put your JavaScript in here, or include an external file.
    let firstDiv = document.getElementsByTagName("div")[0]; // captures the first div of an array
    firstDiv.innerHTML = "Hola Mundo!";

</script>
```

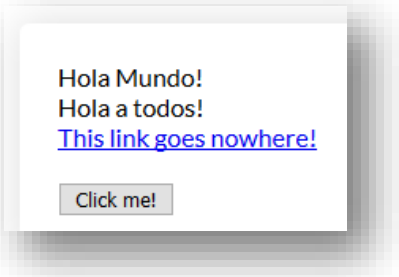


- This can also be done by using the “innerText” property

```
<script>
    // you can put your JavaScript in here, or include an external file.
    let firstDiv = document.getElementsByTagName("div")[0]; // captures the first div of an array
    firstDiv.innerHTML = "Hola Mundo!";

    let secondDiv = document.getElementsByTagName("div")[1];  // captures the second div of an array
    secondDiv.innerText = "Hola a todos!";

</script>
```

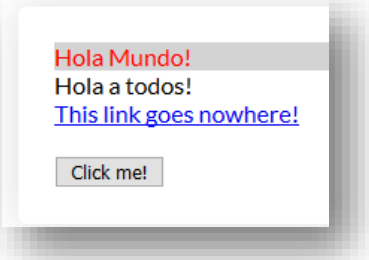


What is the difference between innerText and innerHTML? innerText returns plain text, whereas, innerHTML returns text with formatting.

- Directly manipulate the css properties for elements using the style property

```
<script>
    // you can put your JavaScript in here, or include an external file.
    let firstDiv = document.getElementsByTagName("div")[0]; // captures the first div of an array
    firstDiv.innerHTML = "Hola Mundo!";
    firstDiv.style.color = "#ff0000";
    firstDiv.style.backgroundColor = "#d3d3d3";

    let secondDiv = document.getElementsByTagName("div")[1];  // captures the second div of an arra
```



- Create Elements
    - To create elements use the .createElement function on the document object and pass in a string with the name of the element that we would like to create
    - This will just return a new HTML element without any text/attributes or placement on the page
- Append Elements

```
26          let secondDiv = document.getElementsByTagName("div")[1];  // captures the second div of an array
27          secondDiv.innerText = "Hola a todos!";
28
29          let newDiv = document.createElement("div"); // create a new element
30          let button = document.createElement("button");
31          button.innerText = "I am a button created with JavaScript!";
32          let wrapper = document.getElementById("wrapper");
33          wrapper.appendChild(button);
34
```

- Remove Elements

```
38
39          let removeLink = document.getElementsByTagName("a")[0];  //  remove elements
40          let linkDiv = document.getElementsByTagName("div")[2];
41          linkDiv.removeChild(removeLink);
42
```

- Change Multiple Elements

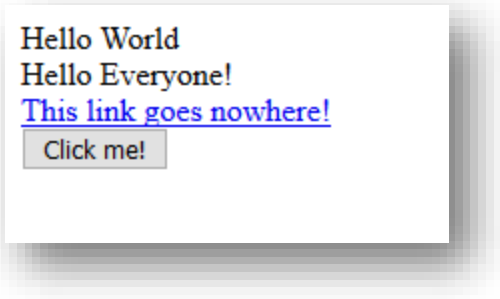
```
        linkDiv.removeChild(removeLink);

        let divs = document.querySelectorAll("div");
        // divs.style.backgroundColor = "#f9f"; // this will not work, try to understand the error you receive!

        // we have to use a loop for each one instead.
        for (let i = 0; i < divs.length; i++)
        {
            divs[i].style.backgroundColor = "#f9f"; // this will work!
        }

    </script>
</body>
```

Start File Result ~



Sample Solution ~ Feel free to add your own flair!

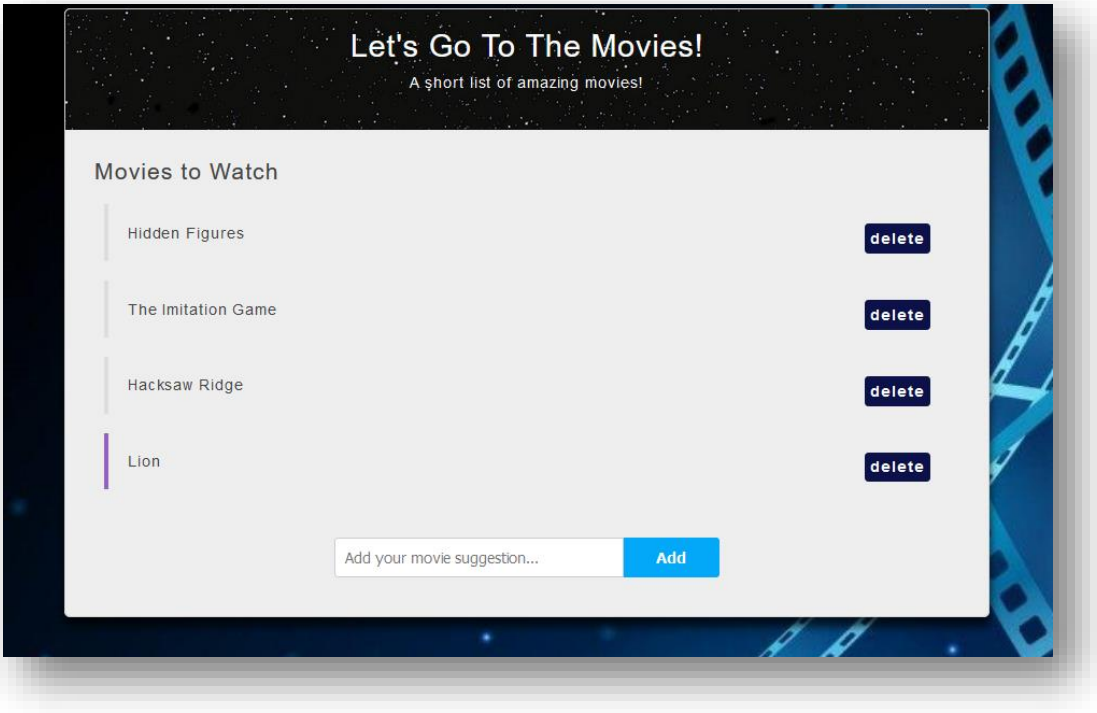
Another great Javascript resource ~

https://www.freecodecamp.org/news/javascript-events-explained-in-simple-english/

Now that you have a foundation for DOM concepts. In this final activity you will create a functional webpage that lists popular movies. In this application, a few DOM manipulation methods are incorporated that makes the webpage interactive.

This will be the only code that should be submitted for credit.

The webpage allows a visitor to delete movies and add their own movie choices.



Launch the “start files” for this activity.

Explore the code:

The HTML file is pretty basic ~ please let me know if there is anything that doesn't make sense with the following code:

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

  <head>
    <!--Wk 7 Lab Final Activity-->

    <meta charset="utf-8">
    <link href="css/styles.css" rel="stylesheet">
    <script src="js/app.js"></script>
    <title>Let's Go to the Movies!</title>

  </head>

  <body>
    <div id="wrapper">
        <header>
            <div id="page-banner">
                <h1 class="title" style="color: #fff;">Let's Go To The Movies!</h1>
                <p style="color: #fff;">A short list of amazing movies!</p><br>
            </div>
        </header>

        <div id="movie-list">
            <h2 class="title">Movies to Watch</h2>
            <ul>

                <li>
                    <span class="name">Hidden Figures</span>
                    <span class="delete">delete</span>
                </li>

                <li>
                    <span class="name">The Imitation Game</span>
                    <span class="delete">delete</span>
                </li>

                <li>
                    <span class="name">Hacksaw Ridge</span>
                    <span class="delete">delete</span>
                </li>

                <li>
                    <span class="name">Lion</span>
                    <span class="delete">delete</span>
                </li>
            </ul>

        </div>

        <form id="add-movie">
            <input type="text" placeholder="Add your movie suggestion...">
            <button>Add</button>
        </form>

    </div>
  </body>
</html>
```

HTML5 standard template which calls a javascript file to perform DOM manipulation. The css file is used to make the webpage attractive.

Obviously, the list items are targeted for modification.

The CSS code:

I am pretty certain that you are comfortable with the various declarations. Please let me know if something doesn't make sense with the code.

```css
body
{
  background-image: url("../images/bg.jpg");
  background-size: 100%;
  font-family: sans-serif;
  color: #444;
  letter-spacing: 1px;
}

h1,h2
{
  font-weight: normal;
}

#wrapper
{
  background-color: #eee;
  width: 90%;
  max-width: 960px;
  margin: 20px auto;
  border-radius: 6px;
  box-shadow: 1px 10px 10px rgba(0, 0, 0, 0.7);
  box-sizing: border-box;
  padding: 0 0 20px;
  overflow: hidden;
  border: 1px solid #d3d3d3;
}

#page-banner
{
  background-image: url("../images/bnr.jpg");
  background-size: 100%;
  align-items: center;
  padding: 10px 0;
}
```

```css
#page-banner h1,#page-banner p
{
  width: 100%;
  text-align: center;
  margin: 10px 0;
}

#movie-list, #add-movie
{
  margin: 30px;
}

#movie-list ul
{
  list-style-type: none;
  padding: 0;
}

#movie-list li
{
  padding: 20px;
  border-left: 5px solid #ddd;
  margin: 20px 10px;
}

#movie-list li:hover
{
   border-color: #9361bf;
}

.delete
{
  float: right;
  background: #0c1249;
  padding: 6px;
  font-weight: bold;
  border-radius: 4px;
  cursor: pointer;
  color: #fff;
}
```

```css

.delete:hover
{
  background: #149cc7;
}

#add-movie
{
  width: 400px;
  margin: 0 auto;
}

#add-movie input
{
  display: block;
  margin: 20px 0;
  padding: 10px;
  border: 1px solid #ccc;
  font-size: 16px;
  border-radius: 4px 0 0 4px;
  box-sizing: border-box;
  width: 300px;
  float: left;
  clear: both;
}


#add-movie button
{
  border: 1px solid #03a8f9;
  background: #03a8f9;
  padding: 10px 20px;
  font-size: 16px;
  font-weight: bold;
  display: inline-block;
  margin: 0;
  border-radius: 0 4px 4px 0;
  cursor: pointer;
  width: 100px;
  float: left;
  margin: 20px 0;
  border-left: 0;
  color: #fff;
}
```

```css

#add-movie button:hover
{
  background: #76c1e6;
}


#add-movie:after
{
  content: "";
  display: block;
  clear: both;
}


#add-movie label
{
  line-height: 52px;
}
```

Finally the JS code that makes the page “come alive!”

- Create a new file and name it app.js; save it inside a js folder/directory
- Code the following ~

```
// load the contents of the webpage
document.addEventListener('DOMContentLoaded', function()
{

    const list = document.querySelector('#movie-list ul');
    const forms = document.forms;

```

```


    // delete movies

    list.addEventListener('click', (e) =>
    {
      if(e.target.className == 'delete')
      {
        const li = e.target.parentElement;
        li.parentNode.removeChild(li);
      }
    });
```

```
19
20
21
22      // add movies
23
24      const addForm = forms['add-movie'];
25      addForm.addEventListener('submit', function(e)
26      {
27        e.preventDefault();
28
29        // create elements
30
31        const value = addForm.querySelector('input[type="text"]').value;
32        const li = document.createElement('li');
33        const movieName = document.createElement('span');
34        const deleteBtn = document.createElement('span');
35
36
37
38        // add text content
39        movieName.textContent = value;
40        deleteBtn.textContent = 'delete';
41
42
43
44        // add classes
45
46        movieName.classList.add('name');
47        deleteBtn.classList.add('delete');
48
49
50
51        // append to DOM
52
53        li.appendChild(movieName);
54        li.appendChild(deleteBtn);
55        list.appendChild(li);
56
57      });
58  })
```

That's a wrap. Please make sure that your page functions as desired. Let me know if you need ANY clarification with any of the code.

Prior to submission, make changes to the webpage/app to make it your own; in other words, the submission must not exactly resemble my movie app. Possibly, make the theme video games as opposed to movies, or songs, whatever school appropriate theme you desire. Modify the CSS to suit your preferences as well!

Deliverables:

- Include a CSS file so that the lab is presented professionally
- Include at least one additional DOM manipulation, include a comment within the code
- Compress ALL of the necessary files for this assignment into one file and then upload to the corresponding drop box
- There will be one (1) drop box for the compressed files