

---

<Semester II>

# <//Programming for Interactive Interfaces//>

Week 8

---

# HTML DOM Methods

- ❑ The HTML DOM can be accessed with JavaScript (and with other programming languages).
- ❑ In the DOM, all HTML elements are defined as objects.
- ❑ The API is the properties and methods of each object.
- ❑ A property is a value that you can get or set (like changing the content of an HTML element).
- ❑ A method is a type of functions that allows you to take some action on your webpage (like add or deleting an HTML element).
- ❑ For example, you can use DOM methods to access HTML elements using id, attribute, tag name, class name, etc., add events to the document or HTML elements, add new HTML elements to the DOM, etc.

# DOM HTMLCollection

- ❑ An **HTMLCollection** is an **array-like collection (list)** of **HTML elements**. (but it's not an array so arrays methods won't work)
- ❑ The elements in a collection can be accessed by index (starts at 0).
- ❑ The **length** Property returns the number of elements in the collection.
- ❑ The **getElementsByTagName()** Method and **getElementsByClassName()** Method - these are two methods that return such an HTML Collection
- ❑ You can loop through the list and refer to the elements with a number (just like an array). Typecast HTMLCollection to an array to be able to use array methods like **forEach()**, **from()**, etc.
- ❑ The following properties and methods can be used on an HTMLCollection: **.length**, **item()**, **namedItem()**.

# document.getElementsByTagName()

- ❑ The `document.getElementsByTagName()` method returns all the element of specified tag name.
- ❑ Syntax: `document.getElementsByTagName("HTMLtag")`
- ❑ Example: `let allparas = document.getElementsByTagName("p");`
- ❑ The `getElementsByTagName()` method returns a collection of all child elements with a given tag name.
- ❑ An `HTMLCollection` is an array-like collection (list) of HTML elements.
- ❑ The `length` Property returns the number of elements in the collection.
- ❑ The elements can be accessed by index (starts at 0).
- ❑ An `HTMLCollection` is live. It is automatically updated when the document is changed.

# document.getElementsByClassName()

- ❑ The `getElementsByClassName()` method returns a collection of elements with a specified class name(s).
- ❑ The `getElementsByClassName()` method returns an `HTMLCollection`.
- ❑ Syntax: `getElementsByClassName("classname");`
- ❑ Eg1: `let infoparas = document.getElementsByClassName("infopara");`  
This will return all the elements that have the class name “infopara”
- ❑ Eg2: `document.getElementsByClassName("red test");`  
This will return all the elements that have both class names “red” and “test”
- ❑ Eg3: `document.getElementById("main").getElementsByClassName("test");`  
Get all elements that have a class of “test”, inside of an element that has the ID of “main”. So those elements will be children of `<main>`

# Accessing a style object

- ❑ The Style object represents an individual style statement.
- ❑ The HTML DOM allows JavaScript to change the style of HTML elements. Minimize the use of DOM to change styles.
- ❑ Syntax: `document.getElementById(id).style.property = new style;`
- ❑ Eg. `document.getElementById("mypara").style.color = "rgb(255, 255, 255)";` // this changes the inline style of the element
- ❑ Almost all of the CSS properties that exist can be modified through JavaScript using the style object.
- ❑ Modifying CSS through Javascript does not impact the Document Object Model (DOM) since it deals with stylistic modifications on existing web page elements. Any HTML element is targetable through Javascript CSS.

# document.querySelector() DOM Method

- ❑ The DOM method `querySelector()` returns the first Element within the document that matches the specified CSS selector, or group of CSS selectors. If no matches are found, null is returned.
- ❑ To return all matches (not only the first), use the `querySelectorAll()` instead.
- ❑ The CSS Selector can be: an HTML tag, CSS class or id
- ❑ Syntax: `document.querySelector("CSS Selector");`  
Eg1: `document.querySelector("img");`  
Eg2: `document.querySelector(".myclass");`  
Eg3: `document.querySelector("#myid");`
- ❑ CSS pseudo-elements will never return any elements.

# document.querySelectorAll()

## DOM Traversal Method

- ❑ The `querySelectorAll()` method returns all elements that matches a CSS selector(s).
- ❑ It returns a `NodeList` which is like an array/ HTML Collection.
- ❑ `querySelectorAll()` returns a static (not live) `NodeList` representing a list of the document's elements that match the specified group of selectors.
- ❑ Syntax: `querySelectorAll(selectors)`
- ❑ Example: `const matches = document.querySelectorAll("p");`
- ❑ The elements are in document order – that is, parents before children, earlier siblings before later siblings.





**Homework: Continue the HTML + CSS portion of your LCA2 project. If design is still left, complete it and create a working prototype/storyboard.**

**W3Schools JavaScript Tutorial**  
**JavaScript for Web | MDN**

# Reference



1. [JavaScript HTML DOM Methods - W3Schools](#)
2. [DOM HTMLCollection Object - W3Schools](#)
3. [HTMLCollection | MDN](#)
4. [getElementsByTagName - Tpoint Tech](#)
5. [HTML DOM Element getElementsByTagName\(\) Method - W3Schools](#)
6. [HTML DOM Document getElementsByClassName\(\) Method](#)
7. [HTML DOM Style object - W3Schools \(List of CSS properties that can be modified with the DOM\)](#)
8. [Changing Element Styling with Javascript CSS | Udacity](#)
9. [Enhanced for loop for HTMLCollection Elements - Stack Overflow](#)
10. [HTML DOM Document querySelector\(\) Method - W3Schools](#)
11. [JavaScript HTML DOM NodeList Object - W3Schools](#)
12. [querySelector\(\) and querySelectorAll\(\) DOM Methods JavaScript - FreeCodeCamp](#)

# Fonts & colors used

This presentation has been made using the following fonts:

## **Blinker**

(<https://fonts.google.com/specimen/Blinker>)

## **Inconsolata**

(<https://fonts.google.com/specimen/Inconsolata>)

#ffffff

#022a46

#72ffdd

#72fff

#0d3a58