ASSIGNMENT 4

Question 1:

By how many ways we can access elements in the DOM and write about them in brief.

Answer :

The methods by which we can access elements in the DOM are:-

## .getElementsByTagName( )

This method will return an array of all the elements you specify between the parentheses.

<html>  
 <p>Accessed by Tag</p>  
 <p>Accessed by Tag</p>  
</html><script> getByTag = document.getElementsByTagName("p");  
</script>

In the example above we are selecting all the paragraph tags in the document and set it equal to a the variable getByTag. Also keep in mind that we are calling this method on the document object.

<html>  
 <button>Click Me</button>  
 <button>Click Me</button>  
</html><script>  
getAllButtons = document.getElementsByTagName("button");  
</script>

This second example will select all of the buttons in the document. As you can observe once again, this method is being called on the document object.

## .getElementsByClassName( )

This method uses the class attribute of an element in order to gain access to it. This allows you to be more specific than .getElementsByTagName() because it will only return the elements that contain the class name you specify within the parentheses.

<html>  
 <button class="hello">Hello</button>  
 <button>Good Morning</button>  
 <button class="hello">Hello</button>  
 <button>Click</button>  
</html><script>  
className = document.getElementsByClassName(".hello");  
</script>

By using this method I will only have access to the elements that have the class attribute of “hello”. This is very helpful if you want to group elements togethers and manipulate those elements as well, without affecting the other elements.

## .getElementById( )

This method is perfect if you want to target one specific element. It uses the id attribute of an element in order to access that element. Beware the id must be unique.

<html>  
 <button>Click Me</button>  
 <button id="myButton">Click Me Also</button>  
 <button>Click Me Again</button>  
</html><script>  
accessById = document.getElementById("myButton");</script>

The code within the script tags will allow me to have access only to the button that has the id of “myButton”. When it comes to specificity this method is the best among the ones I have already discussed.

The following two methods have gain a lot of popularity recently and both use [CSS selectors](https://www.w3schools.com/cssref/css_selectors.asp" \t "https://medium.com/@ralph1786/_blank) in order to gain access to DOM elements.

## .querySelectorAll( )

We can use this method to collect all of the elements that match a specific query. This query is specified within the parentheses.

<html>  
 <p class="query-all">Hello World</p>  
 <p>Good Morning</p>  
 <p class="query-all">How is your day?</p>  
</html><script>  
queryAll = document.querySelectorAll(".query-all");</script>

The example above will result in the method returning an array of all the elements that have the class attribute that we specified.

## .querySelector( )

In a way this method is the counterpart to the previous method discussed. This method returns the first element within the document object that satisfies the CSS selector you provide within the parentheses.

<html>  
 <button>Click Me</button>  
 <button id="only-me">Click Me Also</button>  
 <button>Purchase</button>  
 <button class="back">Go Back</button>  
</html><script>  
onlyMe = document.querySelector("#only-me");  
goBack = document.querySelector(".back");</script>

Question 2:

Write a javascript program that will take 2 numbers from the HTML page and display them on the HTML page after the addition of those 2 numbers.

**Answer:**

// store input numbers

const num1 = parseInt(prompt('Enter the first number '));

const num2 = parseInt(prompt('Enter the second number '));

//add two numbers

const sum = num1 + num2;

// display the sum

console.log(`The sum of ${num1} and ${num2} is ${sum}`);