# LAB 8

In Lab7, JavaScript is used on the server side to dynamically change the content of the web page (edit the HTML). In this lab, we will explore how JavaScript can be used on the client side to make web pages more dynamic and perform programming tasks without the need for you to code your own server-side processing. (We will consider the client as the application sending HTTP request messages and the server as the application sending HTTP response messages.) Web browsers are clients that are typically able to run JavaScript. Javascript code can be inserted into HTML directly <script>alert('123')</script> or placed in an external file <script src="myScript.js"></script>. When the browser receives the html page and files associated with the page, it represents the elements of the web page as objects. This is what is referred to as the document object model (DOM). A complete reference can be found [here](https://www.google.com/url?q=https://www.w3schools.com/js/js_htmldom.asp&sa=D&ust=1490816105324000&usg=AFQjCNH98Pum56fC2-FLPN9WzWcXovuFzg). JavaScript and other programming languages can make changes to these object.  You can think of each element as an object with properties and methods. The innerHTML property allows you to change the content of the element and the getElementId property allows you to access the HTML element by name from the document object (which stores all other HTML element objects for a particular page).

<html>

<body>  
<h1 id="aname" onclick="this.innerHTML = 'You clicked the title!'">Title</h1>

<script>

function modifyHeader()

{  
   document.getElementById("aname").innerHTML = "Change the title name with JavaScript";

}

</script>  
</body>

</html>

With each element, JavaScript can associate a function or event handler to respond to [events](https://www.google.com/url?q=https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events&sa=D&ust=1490816105333000&usg=AFQjCNGmyz3FiV84WZQTrzi2IIPBps4z0w) such as user actions like clicking the element content or hovering the mouse over an element. A list of events are [here](https://www.google.com/url?q=https://www.w3schools.com/jsref/dom_obj_event.asp&sa=D&ust=1490816105334000&usg=AFQjCNGWugycDmIOOKeyoq-5cvGU4eUorQ). Events can be assigned using an attribute or inside the script tag.

ACTIVITY 2

1. Modify the above code to change the type of [HTML DOM mouse even](https://www.google.com/url?q=https://www.w3schools.com/jsref/dom_obj_event.asp&sa=D&ust=1490816105336000&usg=AFQjCNEZH3vHXphV9N4L0qbq8TQi2r3aQg)t and change the content string to reflect the name of the type of event you chose. Test it out in your browser. Provide a screenshot of your code and the browser page.

<html>

<body>  
<h1 id="aname" onclick="fClick">I’m here. Click me / Double click me / Hover on me</h1>

<script>

function fClick()

{  
   document.getElementById("aname").innerHTML = "You clicked";

}

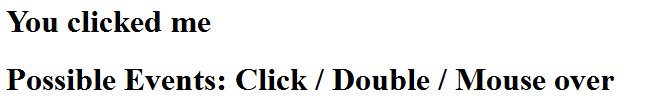
function f1()

{  
   document.getElementById("aname").innerHTML = "You clicked";

}

</script>  
</body>

</html>



**Code is here**

<html>

<body>

<h1 id="aname" onclick="fClick()" ondblclick="fDblClick()" onmouseover="fHover()">

I’m here. Click me / Double click me / Hover on me

</h1>

<h1 >

Possible Events: Click / Double / Mouse over

</h1>

<script>

**function** fClick()

{

document.getElementById("aname").innerHTML = "You clicked me";

}

**function** fDblClick()

{

document.getElementById("aname").innerHTML = "You double clicked me";

}

**function** fHover()

{

document.getElementById("aname").innerHTML = "You are hovering on me";

}

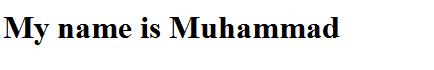
</script>

</body>

</html>

1. Modify the html code so that the onclick event handler is now the modifyHeader function. Assign your name to the innerHTML property. Provide a screenshot of your code and the page displayed in a browser.





<html>

<body>

<h1 id="aname" onclick="modifyHeader()" >

Click me to know my name

</h1>

<script>

**function** modifyHeader()

{

document.getElementById("aname").innerHTML = "My name Muhammad";

}

</script>

</body>

</html>

1. (**20 point bonus/Optional question**) Create an HTML page that has three input text boxes and one button. Add a JavaScript event handler for the button’s onclick event that will translate a word from a source language to target language. Use the XMLHttpRequest code from lab 5. Make sure the request URL contains the values inside the input boxes using document.getElementById("target\_lang").value the value property. Ex. URL: "https://translate.googleapis.com/translate\_a/single?client=gtx&sl=" + document.getElementById("source\_lang").value + "&tl=" + document.getElementById("target\_lang").value + "&dt=t&q=" + document.getElementById("source\_text").value You should use the JavaScript document.write function to display the request’s response.     document.write(req.responseText); Provide a screenshot of before and after clicking the button along with your html code.
2. What does the document.write() function do. Why would you want to avoid using it? Replace this line document.getElementById("aname").innerHTML = "Change the title name with JavaScript"; with document.write("Change the title name with JavaScript"); Provide a screenshot of the displayed page along with your html code

**document.write just write text to html without using any tags. Moreover the place of the text may not be the in the desired locating. That’s why it is better to use document.getElementById("aname").innerHTML, since you have full control of the location of the text.**

1. Read the this [page](https://www.google.com/url?q=https://www.w3schools.com/xml/ajax_intro.asp&sa=D&ust=1490816105342000&usg=AFQjCNF4d5nJV7cPU1ljiOO7z0mheKxG-g) and this [page](https://www.google.com/url?q=https://www.w3schools.com/xml/ajax_xmlhttprequest_send.asp&sa=D&ust=1490816105342000&usg=AFQjCNGdDiUocchNTyxNTeoYWN7_qXtIOg). What is AJAX? (See Lab6) Have we been using the XMLHttpRequest object to send synchronous or asynchronous messages. Why would you want to avoid synchronous requests?

**We have been using the same technique as Ajax does, but we were using synchronous requests. If we set the asynchronous parameter to true, then it would be asynchronous request. We should avoid synchronous requests because JavaScript should wait for the response in synchronous requests, but in asynchronous request browser don’t have to wait for the response. It just checks if status is changed or not.**

1. Another way to add content from a different application is though the iframe element. Iframes allow you to display a page within a page. What does the [youtubeiframeapi](https://www.google.com/url?q=https://developers.google.com/youtube/iframe_api_reference&sa=D&ust=1490816105344000&usg=AFQjCNGyO8ydLWyc53a4PcdLuH5iobGdgw) do? Create an html page using the example code in this link and modify it to display a different video. Change the videoId value.  videoId: 'M7lc1UVf-VE' Make sure the video you have selected has a [creative commons license](https://www.google.com/url?q=https://support.google.com/youtube/answer/2797468?hl%3Den&sa=D&ust=1490816105345000&usg=AFQjCNGQae2id4MIrc_ZC7vUirJRQhmenA). Please provide your code.

**Using IFrame player API you can add a YouTube video player on our website and control the player using JavaScript.**

**<!DOCTYPE html>**

**<html>**

**<body>**

**<!-- 1. The <iframe> (and video player) will replace this <div> tag. -->**

**<div id="player"></div>**

**<script>**

**// 2. This code loads the IFrame Player API code asynchronously.**

**var tag = document.createElement('script');**

**tag.src = "https://www.youtube.com/iframe\_api";**

**var firstScriptTag = document.getElementsByTagName('script')[0];**

**firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);**

**// 3. This function creates an <iframe> (and YouTube player)**

**// after the API code downloads.**

**var player;**

**function onYouTubeIframeAPIReady() {**

**player = new YT.Player('player', {**

**height: '390',**

**width: '640',**

**videoId: 'tp3Gw-oWs2k',**

**events: {**

**'onReady': onPlayerReady,**

**'onStateChange': onPlayerStateChange**

**}**

**});**

**}**

**// 4. The API will call this function when the video player is ready.**

**function onPlayerReady(event) {**

**event.target.playVideo();**

**}**

**// 5. The API calls this function when the player's state changes.**

**// The function indicates that when playing a video (state=1),**

**// the player should play for six seconds and then stop.**

**var done = false;**

**function onPlayerStateChange(event) {**

**if (event.data == YT.PlayerState.PLAYING && !done) {**

**setTimeout(stopVideo, 6000);**

**done = true;**

**}**

**}**

**function stopVideo() {**

**player.stopVideo();**

**}**

**</script>**

**</body>**

**</html>**

1. How would you change the youtubeiframeapi code to not stop after six seconds?

**Just comment this line like below and it will keep playing**

**//setTimeout(stopVideo, 6000);**

1. [Twitter will write your JavaScript and html code for you.](https://www.google.com/url?q=https://publish.twitter.com/%23&sa=D&ust=1490816105347000&usg=AFQjCNF5kTgO-lSw4nEXmJMTZujO27LWpQ) Create a web page and embed a timeline and button. Note you do not need an account to do so. Provide a screenshot of your rendered page and the html code.



<a **class**="twitter-timeline" href="https://twitter.com/aItnps">Tweets by aItnps</a> <script async src="//platform.twitter.com/widgets.js" charset="utf-**8**"></script>

1. Websites can use the [Navigator object](https://www.google.com/url?q=https://www.w3schools.com/jsref/obj_navigator.asp&sa=D&ust=1490816105348000&usg=AFQjCNFZE4HhIqPdPQx6kJhPHHBL-ZWkfg)  to actually track your physical location. Open this [demo](https://www.google.com/url?q=https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/Using_geolocation&sa=D&ust=1490816105349000&usg=AFQjCNGjk6I6frCYTHSFZXpIullcsAjomw) and click the ‘show my location’ button. Take a screenshot of the output. Bowsers and apps determine client location differently. How does the [google maps geolocation api](https://www.google.com/url?q=https://developers.google.com/maps/documentation/geolocation/intro&sa=D&ust=1490816105350000&usg=AFQjCNFSbm5OOupCuwn8iI4eDJ8JClNwog) determine a user’s location?

**Browsers determine location based on the internet provider information. Actually it finds provider location. But cell phones locate you using cell phone company antennas and calculate you latitude and longitude using their methods. The point is cell phone location is exact location.**

 <!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>ACTIVITY 2.9</title>

<meta name="description" content="JavaScript Course">

<script>

**function** geoFindMe() {

**var** output = document.getElementById("out");

**if** (!navigator.geolocation){

output.innerHTML = "<p>Geolocation is not supported by your browser</p>";

**return**;

}

**function** success(position) {

**var** latitude = position.coords.latitude;

**var** longitude = position.coords.longitude;

output.innerHTML = '<p>Latitude is ' + latitude + '° <br>Longitude is ' + longitude + '°</p>';

**var** img = **new** Image();

img.src = "https://maps.googleapis.com/maps/api/staticmap?center=" + latitude + "," + longitude + "&zoom=13&size=300x300&sensor=false";

output.appendChild(img);

}

**function** error() {

output.innerHTML = "Unable to retrieve your location";

}

output.innerHTML = "<p>Locating…</p>";

navigator.geolocation.getCurrentPosition(success, error);

}

</script>

</head>

<body>

<p><button onclick="geoFindMe()">Show my location</button></p>

<div id="out"></div>

</body>

</html>