Week 06 – jQuery

\*A JavaScript library that simplifies JavaScript programming

\*\*Takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

\*\*\*Simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

* HTML/DOM manipulation
* CSS manipulation
* HTML event methods
* Effects and animations
* AJAX
* Utilities
* Also has plugins for almost any task

**INCLUDING IT IN YOUR WEB PAGES:**

Download jQuery library from jQuery.com:

* The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (inside the <head> tag)

<script src="jquery-3.2.1.min.js"></script>

Include jQuery from a CDN ( Content Delivery Network ):

* Google: <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
* Microsoft:

<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.min.js"></script>

When adding JQuery to webpage:

* Can be put either in the head or at the bottom of the body
* If put in the head – must use document.ready() function
* When adding link – ALWAYS PUT AFTER JS? Look up\*\*

**SYNTAX:** made for selecting HTML elements and performing some action on them

* Basic syntax is:  **$(*selector*).*action*()**
* A $ sign to define/access jQuery
* A (*selector*) to "query (or find)" HTML elements
* A jQuery *action*() to be performed on the element(s)
* \*\*Examples:
* $(this).hide() - hides the current element.
* $("p").hide() - hides all <p> elements.
* $(".test").hide() - hides all elements with class="test".
* $("#test").hide() - hides the element with id="test".

Printing Information:

* alert() – prints to alert pop up box
* console.log() – prints to console

Document Ready Event: runs jQuery code when the page is finished loading

* $(document).ready(function()

{

// jQuery methods

});

**JQUERY SELECTORS:**

* Always start with ‘ $() ’
* Used to find/select HTML elements based on their name, id (#), class(.), types, attributes..
* Are based on existing CSS selectors & also has its own custom selectors

$(“p”) 🡪 Selects all <p> elements

**JQUERY EVENT METHODS:**

* jQuery is made to respond to events in an HTML page
* An Event represents the precise moment when something happens
* Most DOM events have an equivalent jQuery method
* Syntax:
* Assign the event to element(s) 🡪 $(“p”).click();

Pass a function defining what should happen when event fires 🡪 $(“p”).click(function()

{

// action goes here

}

Commonly Used jQuery Event Methods:

* Each attaches an event handler function to an HTML form field.
* click(), dblclick()
* mouseenter(), mouseleave()
* mousedown(), mouseup()
* blur(), focus()
* hover()
* on() – attaches one or more event handlers to the selected elements

**JQUERY EFFECTS:**

* HIDE()/SHOW():
* $(selector).hide / .show(speed, callback);
* TOGGLE():
* $(selector).toggle(speed, callback);

Optional Parameters:

* Speed: speed of hiding/showing; “slow”, “fast”, or milliseconds
* Callback: a function to be executed after the method completes
* \*\* $(‘#btn1’).click(function(e){

alert(e.currentTarget.id);

alert(e.currentTarget.innerHTML);

});

* FADE:

Fading methods:

* fadeIn()
* fadeout()
* fadeToggle()
* fadeTo()
* SLIDE:

Sliding methods:

* slideDown()
* slideUp()
* slideToggle()
* ANIMATE:
* animate() method lets you create custom animations

-$(selector).animate({params}, speed, callback);

Required ‘params’ parameter: defined the CSS properties to be animated

* Can Manipulate Multiple Properties at the same time:

\*All property names must be camel-cased while using this method 🡪 paddingLeft

\*\*Color animation not included in jQuery library 🡪 need the ‘Color Animations Plugin’

* Can Define Relative Values:

-Using += or -= in front of value 🡪 the value is then relative to the element’s current value

* Using Pre-defined Values:

-Specify a property’s animation value as “show”, “hide”, or “toggle”

* Default Queue Functionality:

-When you want to perform multiple different animations after each other, jQuery creates an internal queue with method calls & runs the animate calls one by one

* STOP: stops animation/effect before it is finished
* $(selector).stop(stopAll, goToEnd);
* Optional Parameters:

-stopAll: specifies whether the animation queue should be cleared or not

Default: false 🡪 only the active animation will stop, any queued animations perform after

-goToEnd: specifies whether or not to complete the current animation immediately

Default: false

* CALLBACK FUNCTION:

Executed after the current effect is 100% finished

Prevents the next line of code being run when the effect is not finished

* $(selector).hide(speed, callback)

$("button").click(function(){  
    $("p").hide("slow", function(){  
        alert("The paragraph is now hidden");  
    });  
});

// \*\*the callback parameter is a function that executes after the hide effect is complete

METHOD CHAINING:

* Allows us to run multiple jQuery methods, on the same element(s), in a single statement

$("#p1").css("color", "red").slideUp(2000).slideDown(2000);

// chains together the css(), slideUp(), and slideDown() methods,

‘ p1 ’ element first changes to red, then it slides up, and then it slides down

**JQUERY DOM MANIPULATION: \*\***jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate HTML elements and attributes

GET:

* Text() - Sets or returns the text content of selected elements $("#btn1").click(function(){  
      alert("Text: " + $("#test").text()); // Gets Content  
  });
* html() - Sets or returns the content of selected elements (including HTML markup) $("#btn2").click(function(){  
      alert("HTML: " + $("#test").html()); // Gets Content  
  });
* val() - Sets or returns the value of form fields $("#btn1").click(function(){  
      alert("Value: " + $("#test").val()); // Gets Value of Input Field  
  });
* attr() – used to get attribute values

$("button").click(function(){  
    alert($("#w3s").attr("href"));  
}); // Gets Value of the href Attribute in a link

SET:

$("#btn1").click(function(){  
    $("#test1").text("Hello world!");   
});  
 $("#btn2").click(function(){ // Same 3 methods  
    $("#test2").html("<b>Hello world!</b>"); sets content  
});  
 $("#btn3").click(function(){  
    $("#test3").val("Dolly Duck");  
});

$("button").click(function(){  
    $("#w3s").attr("href", "https://www.w3schools.com/jquery");   
}); // Sets Value of href Attribute in a link

* Callback Function: for these 4 methods

-2 parameters: The **index of the current element** in the list of elements selected and the original **(old) attribute value**. You then return the string you wish to use as the new value from the function

ADD:

* append() - Inserts content at the end of the selected elements
* $("p").append("Some appended text.");
* prepend() - Inserts content at the beginning of the selected elements
* $("p").prepend("Some prepended text.");
* Adding several new elements with append() & prepend()

function appendText() {  
    var txt1 = "<p>Text.</p>";               // Create element with HTML    
    var txt2 = $("<p></p>").text("Text.");   // Create with jQuery  
    var txt3 = document.createElement("p");  // Create with DOM  
    txt3.innerHTML = "Text.";  
    $("body").append(txt1, txt2, txt3);      // Append the new elements   
}

* after() - Inserts content after the selected elements
* $("img").after("Some text after");
* before() - Inserts content before the selected elements
* $("img").before("Some text before");
* Adding several new elements with after() & before()

function afterText() {  
    var txt1 = "<b>I </b>";                   // Create element with HTML    
    var txt2 = $("<i></i>").text("love ");    // Create with jQuery  
    var txt3 = document.createElement("b");   // Create with DOM  
    txt3.innerHTML = "jQuery!";  
    $("img").after(txt1, txt2, txt3);         // Insert new elements after

<img>  
}

REMOVE:

* remove() - Removes the selected element (and its child elements)
* $("#div1").remove();
* Also accepts one parameter that allows you to filter the element(s) to be removed
* $("p").remove(".test"); $("p").remove(".test, .demo");
* empty() - Removes the child elements from the selected element
* $("#div1").empty();

**CSS MANIPULATION:** GET & SET CSS Classes

* addClass() - Adds one or more classes to the selected elements
* removeClass() - Removes one or more classes from the selected elements
* toggleClass() - Toggles between adding/removing classes from the selected elements
* css() - Sets or returns one or more style properties for the selected elements
* Return a CSS Property: css(“propertyname”);

$("p").css("background-color");

* Set a CSS Property: css(“*propertyname*”, “*value*”);

$("p").css("background-color", "yellow");

* Set multiple CSS Properties: css({“*propertyname*”: “*value*","*propertyname*”: “*value*”,...});

$("p").css({"background-color": "yellow", "font-size": "200%"});

**JQUERY DIMENSIONS:**

jQuery has several important methods for working with dimensions:

* width()
* height()
* innerWidth()
* innerHeight()
* outerWidth()
* outerHeight()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*References: on W3Schools\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* jQuery Selectors
* jQuery Events (event methods)
* jQuery Effects (effect methods)
* jQuery HTML/CSS (methods for manipulating)
* jQuery Properties

jQuery Video Lecture Notes

**Linking jQuery into your code:**

* Google hosted libraries 🡪 developers.google.com

- jQuery:

src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">

-jQuery UI:

<scriptsrc="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jqueryui.min.j></script>

* jQuery.api.com 🡪 entire api and methods
* jQuery Selectors: does the query
* jQuery Methods: on result of the query it performs actions

**Assignment:**

var x = $(“Test”); 🡪 wrapping the query inside the dollar sign turns this into a jQuery object; can now invoke all of the jQuery methods on it

Variables Declared:

* isHoldingPiece = false 🡪 you want to know if you’re currently holding a piece or not
* currentColor = “”; 🡪 you want to know what color the piece is currently
* lastCell = 🡪 you want to know what the last cell you clicked on was

Switching Colors: (Red to Black)

var allBlackPieces = $(“.black”);🡪returns all elements w/ ‘black’ as or anywhere in the class name

allBlackPieces.removeClass(“black”);

allBlackPieces.addClass(“red”);

**Tips:**

* **$** means jQuery; Must Use “.” w/ class name
* **Shift Refresh** clears cashed requests from browser
* Store ‘$(this)’ in a variable
* When in doubt wrap in ‘$( )’
* Might Be Useful:

var everyElementOnThePage = $(“div”);

everyElementOnThePage.Click(function() 🡪 performs on ~ every element

{

});

$(".piece").click(function ()

{

if (($(this).attr("class") == "cell piece red") || ($(this).attr("class") == "cell piece black"))

{

$(this).addClass("selected");

$(this).removeClass("cell");

$(this) = selectedCell;

}

});

**jQuery Donnie videos:**

\*jQuery link allows you to access the jQuery operator 🡪 ‘ $ ’

**JQUERY UI:**

Declare one empty object 🡪 var Home = { }; 🡪 establishes scope

* Attach all functions/variables to that object