**JQUERY BASICS**

**Link to Jquery's Source Code Remotely-**

<script type="text/javascript"src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js">

</script>

Running Code Unobtrusively When the DOM is Ready-

$(document).ready(function() {

// all jQuery code goes here

});

**Select Elements in JQuery-**

$(“div”); // selects all HTML div elements

$(“#myElement”); // selects one HTML element with ID “myElement”

$(“.myClass”); // selects HTML elements with class “myClass”

$(“p#myElement”); // selects paragraph elements with ID “myElement”

$(“ul li a.navigation”); // selects anchors with class “navigation” that are nested in list items

**JQuery Supports all the CSS Selectors**

$(“p > a”); // selects anchors that are direct children of paragraphs

$(“input[type=text]”); // selects inputs that have specified type

$(“a:first”); // selects the first anchor on the page

$(“p:odd”); // selects all odd numbered paragraphs

$(“li:first-child”); // every list item that’s first child in a list

**Jquery Also AllowS the Custom Selectors-**

$(":animated"); // selects elements currently being animated

$(":button"); // selects any button elements (inputs or buttons)

$(":radio"); // selects radio buttons

$(":checkbox"); // selects checkboxes

$(":checked"); // selects selected checkboxes or radio buttons

$(":header"); // selects header elements (h1, h2, h3, etc.)

**Manipulating and Accessing CSS Class Names-**

$("div").addClass("content"); // adds class "content" to all <div> elements

$("div").removeClass("content"); // removes class "content" from all <div> elements

$("div").toggleClass("content"); // toggles the class "content" on all <div> elements (adds it if it doesn't exist, and removes it if it does)

You can also check to see if a selected element has a particular CSS class, and then run some code if it does. You would check this using an if statement. Here is an example:

if ($("#myElement").hasClass("content")) {

// do something here

}

**Manipulating CSS Styles with jQuery**

$("p").css("width", "400px"); // adds a width to all paragraphs

$("#myElement").css("color", "blue") // makes text color blue on element #myElement

$("ul").css("border", "solid 1px #ccc") // adds a border to all lists

**Adding, Removing, and Appending Elements and Content**

Get the HTML of any element (similar to innerHTML in JavaScript):

let myElementHTML = $("#myElement").html(); // variable contains all HTML (including text) inside #myElement

If you don’t want to access the HTML, but only want the text of an element:

let myElementHTML = $("#myElement").text(); // variable contains all text (excluding HTML) inside #myElement

$("#myElement").html("<p>This is the new content.</p>"); // content inside #myElement will be replaced with that specified

$("#myElement").text("This is the new content."); // text content will be replaced with that specified

**To append content to an element:**

$("#myElement").append("<p>This is the new content.</p>"); // keeps content intact, and adds the new content to the end

$("p").append("<p>This is the new content.</p>"); // add the same content to all paragraphs

jQuery also offers use of the commands appendTo(), prepend(), prependTo(), before(), insertBefore(), after(), insertAfter(), which work similarly to append but with their own unique characteristics

Dealing with Events in jQuery

$("a").click(function() {

// do something here

// when any anchor is clicked

});

The code inside function() will only run when an anchor is clicked. Some other common events you might use in jQuery include blur, focus, hover, keydown, load, mousemove, resize, scroll, submit, select.

**Showing and Hiding Elements with jQuery**

The syntax for showing, hiding an element (or toggling show/hide) is:

$("#myElement").hide("slow", function() {

// do something once the element is hidden

}

$("#myElement").show("fast", function() {

// do something once the element is shown

}

$("#myElement").toggle(1000, function() {

// do something once the element is shown/hidden

}

Remember that the “toggle” command will change whatever state the element currently has, and the parameters are both optional. The first parameter indicates the speed of the showing/hiding. If no speed is set, it will occur instantly, with no animation. A number for “speed” represents the speed in milliseconds. The second parameter is an optional function that will run when the command is finished executing.

**You can also specifically choose to fade an element in or out, which is always done by animation**:

$("#myElement").fadeOut("slow", function() {

// do something when fade out finished

}

$("#myElement").fadeIn("fast", function() {

// do something when fade in finished

}

**To fade an element only partially, either in or out:**

$("#myElement").fadeTo(2000, 0.4, function() {

// do something when fade is finished

}

The second parameter (0.4) represents “opacity”, and is similar to the way opacity is set in CSS. Whatever the opacity is to start with, it will animate (fadeTo) until it reaches the setting specified, at the speed set (2000 milliseconds). The optional function (called a “callback function”) will run when the opacity change is complete. This is the way virtually all callback functions in jQuery work.

**jQuery Animations and Effects**

You can slide elements, animate elements, and even stop animations in mid-sequence. To slide elements up or down:

$("#myElement").slideDown("fast", function() {

// do something when slide down is finished

}

$("#myElement").slideUp("slow", function() {

// do something when slide up is finished

}

$("#myElement").slideToggle(1000, function() {

// do something when slide up/down is finished

}

To animate an element, you do so by telling jQuery the CSS styles that the item should change to. jQuery will set the new styles, but instead of setting them instantly (as CSS or raw JavaScript would do), it does so gradually, animating the effect at the chosen speed:

$("#myElement").animate(

{

opacity: .3,

width: "500px",

height: "700px"

}, 2000, function() {

// optional callback after animation completes

}

);

Animation with jQuery is very powerful, and it does have its quirks (for example, to animate colors, you need a special plugin). It’s worth taking the time to learn to use the animate command in depth, but it is quite easy to use even for beginners.