Murach’s JavaScript and Jquery Chapter 7 jQuery Essentials

jQuery uses CSS selectors to select the HTML elements that the methods should be applied to. For example:

$(“#faqs h2”)

Is a jQuery selector for the CSS selector

#faqs h2

Which select all of the h2 elements in the element with “faqs” as its id. In fact, jQuery supports all of the CSS selectors including the CSS3 selectors.

**jQuery Core**

**How to code jQuery selectors**

When you use jQuery, you start by selecting the element or elements that you want to apply a jQuery method to.

To code a jQuery selector, you start by coding the dollar sign ($) followed by set of parentheses that contains a set of quotation marks. Then, within the quotation marks, you code the CSS selector for the element or elements that you want to select.

**The syntax for a jQuery selector:**

$(“selector”)

**How to select elements by element, id, and class:**

By element type: All <p> elements in the entire document

$(“p”)

By id: The element with “faqs” as its id

$(“#faqs”)

By class: All elements with “plus” as a class

$(“.plus”)

**How to select elements by relationship**

Descendants: All <p> elements that are descendants of the “#faqs” id element

$(“div p”)

Adjacent siblings: All div elements that are placed immediately after h2 elements

$(“h2 + div”)

General siblings: All <p> elements that are preceded by ul element with the same parent

$(“h2 ~ div”)

Children: All ul elements that the parent is div element

$(“div > p”)

**How to code multiple selectors**

$(“#faqs li, div p”)

$(“p + ul, div ~ p”)

NOTE: when you use jQuery, the dollar sign ($) is used to refer to the jQuery library. Then you can code *selector* by using the CSS syntax within quotation marks within parentheses.

**How to call jQuery methods**

Once you have selected the element or elements that you want to apply a *method* to, you will call the method using the syntax below. This is the same way that you call a method of any method of any object. You code the selector that gets the element or elements, the dot, the method name, and any parameters within parentheses.

$(“selector”).methodName(parameters)

|  |  |
| --- | --- |
| Method | Description |
| val() | Get the value of a text box or other form control. |
| val(value) | Set the value of a text box or other form control. |
| text() | Get the text of an element. |
| text(value) | Set the text of an element. |
| next(type) | Get the next sibling of an element or the next sibling of a specified type if the parameter is coded. |
| submit() | Submit the selected form. |
| focus() | Move the focus to the selected form control or link. |

Examples:

**How to get the value from a text box**

var gallons = $(“#gallons”).val();

**How to set the value for an input element**

$(“#gallons”).val(“”);

**How to set the text in an element**

$(“#email\_address\_error”).text(“Email address is required”);

**How to set the text for the next sibling with object chaining**

$(“#last\_name”).next().text(“Last name is required”);

**How to submit a form**

$(“#join\_list”).submit();

**How to move the focus to a form control or link**

$(“#email\_address”).focus();

NOTE: when you use *object chaining* with jQuery, you code one method after the other. This works because each method returns the appropriate object.

If the selector for a method selects more than one element, jQuery applies the method to all of the elements so you don’t have to code a loop to do that.

**How to use jQuery event methods**

When you use jQuery, you use *event methods* to attach event handlers to events. To do this, you use the syntax below. First, you code the selector for the element that will initiate the event like a button that will be clicked. Then, you code the name of the event method that represents the event that you want to use. Last, you code a function that will be the event handler for the event within parentheses.

$(“selector”).eventMethodName(function() {

// the statements of the event handler

});

Two common jQuery event methods

|  |  |
| --- | --- |
| Event method | Description |
| ready(handler) | The event handler runs when the DOM is ready. |
| click(handler) | The event handler runs when the selected element is clicked. |

**Two ways to code an event handler for the jQuery ready event**

**The long way**

$(document).ready(function() {

alert(“The DOM is ready”);

});

**The short way**

$(function() { // (document).ready is assumed

alert(“The DOM is ready”);

});

**An event handler for the click event of all h2 elements**

$(“h2”).click(function() {

alert(“This heading has been clicked”);

});

**The click event handler within the ready event handler**

$(document).ready(function() {

$(“h2”).click(function() {

alert(“This heading has been clicked”);

}); // end of click event handler

}); // end of ready event handler

NOTE: the ready event is the jQuery alternative to the JavaScript load event, except that it works better. Unlike the load event, the ready event is triggered as soon as the DOM is built, even if other elements like images are still being loaded into the browser. This means that the user can start using the web page faster.

When coding one event handler within another, the use of the closing braces, parentheses, and semicolons is critical. To help get this right, we should code inline comments after these punctuation marks to identify the ends of the handlers.

**A working subset of selectors, methods, and event methods**

**The most useful selectors**

A summary of the most useful jQuery selectors

|  |  |
| --- | --- |
| Selector | Selects |
| [attribute] | All elements with the named attribute. |
| [attribute=value] | All elements with the named attribute and value. |
| :contains(text) | All elements that contain the specified text. |
| :empty | All elements with no children including text nodes. |
| :eq(n) | The element at index n within the selected set. |
| :even | All elements with an even index within the selected set. |
| :first | The first element within the set. |
| :first-child | All elements that are first children of their parent elements. |
| :gt(n) | All elements within the selected set that have an index greater than n. |
| :has(selector) | All elements that contain the element specified by the selector. |
| :header | All elements that are headers (h1, h2, …) |
| :hidden | All elements that are hidden. |
| :last | The last element within the selected set. |
| :last-child | All elements that are the last children of their parent elements. |
| :lt(n) | All elements within the selected set that have an index less than n. |
| :not(selector) | All elements that are not selected by the selector. |
| :nth-child | All elements that are the nth children of their parent elements. |
| :odd | All elements with an odd index within the selected set. |
| :only-child | All elements that are the only children of their parent elements. |
| :parent | All elements that are parents of other elements, including text nodes. |
| :text | All input elements with the type attribute set to “text”. |
| :visible | All elements that are visible. |

Examples:

**How to select the li elements that are the first child of their parent element**

$(“li:first-child”)

**How to select the even tr elements of a table**

$(“table > tr:even”) // numbering starts at 0, so first tag is even

**How to select the third descendant <p> element of an element**

$(“#faqs p:eq(2)”) // numbering starts at 0

**How to select all input elements with “text” as the type attribute**

$(“:text”)

**The most useful methods**

A summary of the most useful jQuery methods

|  |  |
| --- | --- |
| Method | Description |
| next([selector]) | Get the next sibling of an element or the first sibling of a specified type if the parameter is coded. |
| prev([selector]) | Get the previous sibling of an element or the previous sibling of a specified type if the parameter is coded. |
| attr(attributeName) | Get the value of the specified attribute. |
| attr(attributeName, value) | Set the value of the specified attribute. |
| css(propertyName) | Get the value of the specified property. |
| css(propertyName, value) | Set the value of the specified property. |
| addClass(className) | Add one or more classes to an element and, if necessary, create the class. If you use more than one class as the parameter, separate them with spaces. |
| removeClass([className]) | Remove one or more classes. If you use more than one class as the parameter, separate them with spaces. |
| toggleClass(className) | If the class is present, remove it. Otherwise, add it. |
| hide([duration]) | Hide the selected element. The duration parameter can be “slow”, “fast”, or a number giving the time in milliseconds. By default, the duration is 400 milliseconds, “slow” is 600 millisecond, and “fast” is 200 milliseconds. |
| show([duration]) | Show the selected element. The duration parameter is the same as for the hide method. |
| each(function) | Run the function for each element in an array. |

Example:

**Get the value of the src attribute of an image**

$(“#image”).attr(“src”);

**Set the value of the src attribute of an image to the value of a variable**

$(“#image”).attr(“src”, imageSource);

**Set the value of the color property of the h2 elements to blue**

$(“h2”).css(“color”, “blue”);

**Add a class to the h2 descendants of the “faqs” element**

$(“#faqs h2”).addClass(“minus”);

**Run a function for each <a> element within an “image\_list” element**

$(“#image\_list a”).each(function() {

// the statements of the function

});

**The most useful event methods**

A summary of the most useful jQuery event methods

|  |  |
| --- | --- |
| Event method | Description |
| ready(handler) | The handler runs when the DOM is ready. |
| unload(handler) | The handler runs when the user closes the browser window. |
| error(handler) | The handler runs when a JavaScript error occurs. |
| click(handler) | The handler runs when the selected element is clicked. |
| toggle(handlerEven, handlerodd) | The first handler runs on even clicks of the element, starting with 0. The second handler runs on odd clicks of the element. |
| dblclick(handler) | The handler runs when the selected element is double clicked. |
| mouseenter(hanlder) | The handler runs when the mouse pointer enters the selected element. |
| mouseover(handler) | The handler runs when the mouse pointer moves over the selected element. |
| mouseout(handler) | The handler runs when the mouse pointer moves out of the selected element. |
| hover(handlerIn, handlerOut) | The first event handler runs when the moue pointer moves into an element. The second event handler runs when the mouse moves out. |

Example:

**A handler for the double-click event of all text boxes that clears the clicked box**

$(“:text”).dblclick(function () {

$(this).val(“”);

});

**A handler for the hover event of each img element within a list**

$(“#image\_list img”).hover(

function () {

alert(“The mouse pointer has moved into an img element”);

},

function () {

alert(“The mouse pointer has moved out of an img element);

}

); // end hover

**NOTE:**  The table above presents the event methods that you will use the most. Not included are: keydown, keypress, keyup, mousedown, mouseleave, and mousemove.

**Other event methods that you should be aware of**

Other event methods that you should be aware of

|  |  |
| --- | --- |
| Event method | Description |
| bind(event, handler) | Attach an event handler to an event. |
| unbind(event, handler) | Remove an event handler from an event. |
| one(event, handler) | Attach an event handler and remove it after it runs one time. |
| trigger(event) | Trigger the event for the selected element. |

**How to store an event handler in a variable**

var clearClick = function () {

// the statements for the event handler

}

**How to attach an event handler to an event**

**with the bind method**

$(“#clear”).bind(click, clearClick);

**with the shortcut method**

$(“#clear”).click(clearClick);

**How to attach an event handler to two different events**

$(“#clear”).click(clearClick);

$(“:text”).dblclick(clearClick);

**How to remove an event handler from an event**

$(“#clear”).unbind(“click”, clearClick);

**How to attach and remove an event handler so it runs only once**

$(“#clear”).one(“click”, confirmClick);

**How to trigger an event**

**with the trigger method**

$(“#clear”).trigger(“click”);

**with the shortcut method**

$(“#clear”).click();

**How to use the shortcut method to trigger an event from an event handler**

$(“:text”).dblclick(function () {

$(“#clear”).click(); // triggers the click event of the clear button

});

**Note:**

1. When you store an event handler in a variable, you can use the bind method to attach it to more than one event.
2. When you use the shortcut method to attach an event handler to an event, you are actually using the bind method.
3. The submit and focus methods are actually event methods. When you use the shortcut method to trigger them, they trigger the submit and focus events.

**How to use effects**

**The jquery methods for effects**

The basic methods for jQuery effects

|  |  |
| --- | --- |
| Method | Description |
| show() | Display the selected elements from the upper left to the lower right. |
| hide() | Hide the selected elements from the lower right to the upper left. |
| toggle() | Display or hide the selected elements. |
| slideDown() | Display the selected elements with a sliding motion. |
| slideUp() | Hide the selected elements with a sliding motion. |
| slideToggle() | Display or hide the selected elements with a sliding motion. |
| fadeIn() | Display the selected elements by fading them in to opaque. |
| fadeOut() | Hide the selected elements by fading them out to transparent. |
| fadeToggle() | Display or hide the selected elements by fading them in or out. |
| fadeTo() | Adjust the opacity property of the selected elements to the opacity set by the second parameter. With this method, the duration parameter must be specified. |

**The basic syntax for all of the methods except the fadeTo method**

methodName([duration][, callback])

**The basic syntax for the fadeTo method**

fadeTo(duration, opacity[, callback])

**HTML for a heading that is animated after the web page is loaded**

<h1 id=”startup\_message”>Temporarily under construction!</h1>

**jQuery that fades the heading out over 5 seconds**

$(“#startup\_message”).fadeOut(5000);

NOTE: Besides the core jQuery library, jQuery provides the jQuery UI (User Interface) library. The functions in this library use the core jQuery library to build advanced features that can be created with just a few lines of code. These features include themes, effects, widgets, and mouse interactions.