# DOM Review

## Objectives:

- Review the DOM
- Recall how to use JavaScript to interact with the DOM
- Review jQuery event listeners and DOM manipulation

---

Can you remember all the way back to Web Fundamentals when you learned about JavaScript, the DOM, and DOM manipulation with JavaScript and jQuery? Here's a brief review.

## Vanilla JavaScript

The DOM is short for **Document Object Model**, simply meaning that a *document*, like an HTML page, is made up of *objects*. On this page, open your browser's console (right click and inspect) and use JavaScript to get an element from the DOM (there's a div element on this page with the id main_navigation):

```
var nav = document.getElementById("main_navigation");
```

So what did we get? What is `nav` ? Use the `typeof` operator to determine its type.

```
console.log(typeof nav)
```

There it is! It's an object! It's an element from the HTML represented as an object! Hence, Document Object Model.

Since nav is an object, we should be able to iterate over its different keys.

```
for(var key in nav) {
        console.log(key);
}
```

You should see a long output of all the different keys that make the `nav` object - properties and methods such as `innerText` , `style` , and `click` . This gives us access to all the different things this element has and can do. The possibilities are endless! We can change whatever we'd like about the elements and make them do different things! In other words, we can **manipulate the DOM**. Go ahead! Give it a try! If you're unsure of what to change, try this:

```
nav.innerText = "I just manipulated the DOM!";
nav.style.backgroundColor = "yellow";
```

Neat, right?!

## jQuery

**jQuery** is just a JavaScript library that simplifies writing scripts for DOM manipulation. To add jQuery to your page, grab the script link from Google Hosted Libraries. Use jQuery's documentation for a full list of all event listeners and methods available. The snippets below provide a quick review of the most common things we'll do with jQuery. Also go back to your jQuery Functionsassignment for your own examples!

### DOM Manipulation

The jQuery library provides lots of different methods, used like so:

$("selector").jquery_method(value)

| Examples: | Examples: | Potentially |
| --- | --- | --- |
| #container (id) | html | optional |
| .xyz (class) | append | |
| div (element) | css | |

For example, to do what we did above:

```
$("#main_navigation").text("I just manipulated the DOM!");
$("#main_navigation").css("backgroundColor", "yellow");
```

### Adding Event Listeners

A subset of important jQuery methods are those that allow us to control *when* DOM manipulation occurs. If we want to do something if the user performs a certain action, we can add an event listener to an element on the DOM and provide a handler, or a function that explains what we want to do when that action is taken.

$("selector").event_listener(handler)

| Examples: | Examples: | a function you |
| --- | --- | --- |
| #container (id) | click | write—what to |
| .xyz (class) | hover | do when the |
| div (element) | submit | *event* happens |

For example, given the following HTML, suppose we updated the div contents when it is clicked on, or the text color of xyz-classed elements change when p tags are hovered over.

**jquery-review/example.html**

**jquery-review/script.js**

```html
<head>
    // import jQuery
    <script src="script.js" defer>
    </script>
</head>

<body>
    <div id="container">
        <p class="xyz">Coding</p>
        <p class="xyz">Dojo</p>
    </div>
</body>
```

```javascript
$("#container").click(function() {
    // add a new paragraph to the container div
        $("#container").append("<p>New paragraph!</p>");
});

$("p").hover(function() {
        // change the text color of all elements with class 'xyz' to blue
        $(".xyz").css("color", "blue");
    },
    function() {
        // change the text color of all elements with class 'xyz' to black
        $(".xyz").css("color", "black");
});
```