Module #5 — Lesson 1

Objective: Students should have an understanding of what the jQuery library is and understand the steps needed to utilize the jQuery library to manipulate the DOM.

Visuals: Describe the type of visuals this video will need and where they're needed. For example, screen capture, talking head, static or motion graphics, or live action b-roll, etc. The second column is also where you can indicate on screen text.

FSFO-758

VO Script	Visuals
Hello, and welcome! In this video, we'll discuss jQuery and its usefulness in interacting with the Document Object Model, or DOM.	Talking head
JavaScript alone is not ideal for performing extensive interactions with the DOM. But jQuery, a library that extends the features of JavaScript, provides a simplified syntax for doing those very tasks.	
But what is a library? A JavaScript library is a pre-built set of JavaScript methods that allow you to more easily perform functions. There are many JavaScript libraries available, but jQuery is one of the most popular ones. Let's take a look at why that is.	Display definition: "A **library** is a pre-built set of JavaScript methods that allow you to more easily perform functions."
jQuery is a small, fast, and feature-rich JavaScript library that allows developers to do things like manipulate the DOM, handle events, and even add animation to pages. jQuery claims to be the "write less, do	Display:

more” library. This means it helps you create more complex functionality with less coding than if you were using JavaScript alone.	“With jQuery, you can write less and do more.”
Let’s look at an example. In this code example, we use JavaScript to identify two elements in the DOM: an element with an ID of `logout` and an element with an ID of `logoutMenu`. Then we add an event listener to the `logout` element. When it’s clicked, we use an `if statement` to check whether the `logoutMenu` is already displayed. If it isn’t, then we display it. If it is, then we hide it. Essentially, we are toggling the logout menu on and off.	Display: ``` var logout = document.querySelector("#logout"); var logoutMenu = document.querySelector("#logoutMenu"); logout.addEventListener("click", function(){ if(logoutMenu.style.display === "none") { logoutMenu.style.display = "inline"; } else { logoutMenu.style.display = "none"; } }); ```
That’s a lot of code, isn’t it? Let’s see how we could do the same thing using jQuery.	
Using jQuery, we can reduce nine lines of code to three. Which one would you rather write? When an application is thousands of lines long, this can make a huge difference.	Display: ``` $("#logout").on(“click”, function(){ $("#logoutMenu").toggle() }); ```
Even more importantly, though, as a full-stack web developer, you are certain to come across **many** applications that use jQuery. It’s important to understand how it works so you can fix bugs and add features to the existing codebase—even if you prefer to use plain old JavaScript.	

That’s all for now. Let’s get back to the lesson!	