* - [Narrator] Let's start at the very beginning with the question, where does the JavaScript live? Where do you actually write the JavaScript code? There are several different approaches to storing and serving JavaScript in an HTML document most prominently inline meaning the JavaScript itself is literally in line inside the document and as an external file .
* In exercise files for this movie 02\_01 you'll find index dot HTML and this is an entirely self-contained HTML document that has both styles and JavaScript inline .
* You can see the styles at the very top here there's a style tag and then some style declarations and then an end style tag .
* Then if we scroll all the way to the bottom of the document right before the end body tag, we have a script tag with some JavaScript inside and then an end script tag .
* This script tag is how you add JavaScript to a document .
* Anything inside the script tag will automatically be interpreted by the browser as JavaScript and rendered as such and the script tag can actually be placed anywhere within the document .
* Now, technically speaking, you would want to place it in the head section because it's a meta part of the document it's not part of the actual document but in many cases, you'll find examples, tutorials even applications that place the script tag at the very end of the document .

Graphical user interface, text

Description automatically generated

* There's a good reason for this and it has to do with how the browser renders content in general, when you give the browser an HTML documents the browser just starts to read it from the top, line by line .
* It renders it out as it reads it and then anytime it encounters JavaScript everything stops that's because the JavaScript may make changes to the document .
* So when the browser encounters a script tag, it says, "Stop all rendering, I need to see what's going on here .
* " And then reads all the JavaScript, renders the JavaScript and then goes back to rendering the entire page .
* That's important because if the JavaScript is trying to do something to something else that's being rendered in the page, it has to be called after the thing has been rendered and in the example here, that's exactly what's happening .
* This piece of code looks for any instance of the code element and you can see there's some code elements in the HTML here, but if we're going to have the browser find the code elements the browser has to have rendered the code elements first and that means the JavaScript needs to appear after the code elements, which is why the script tag is at the bottom of the document .
* Having settled this, what you're seeing here is an anti-pattern .
* This is how we used to do things and this is what you'll see in a lot of tutorials, a lot of code examples and even a lot of documentation .
* However, today we have better tools and better ways of loading JavaScript and controlling what happens .
* So in a couple of movies, we'll cover JavaScript loading the modern way in more detail but for now know that the script tag wraps around any JavaScript and you can place that script tag in the head section, inside the body, outside the body basically anywhere within the HTML element and it'll all work .