* - [Instructor] Writing JavaScript in line in an HTML document like this is a bit of an edge case, and it only really makes sense when you have just a little bit of JavaScript, and it only applies to the current document and nowhere else .
* In most cases, when you write JavaScript, you'll have a lot of script and you want to apply it to several different documents .
* In those circumstances, the best practice is to take the script, place it in its own file or several different files, and then just reference the files whenever you want to use them .
* There's already a model for this in how we handle CSS .
* If you have just a few lines of CSS and you want to apply it to just one document, it's fine to place it inside the document using the style tag, but in most cases, you want to place your styles in a separate style sheet and then call the style sheet in when you want to use it .
* We're going to do the exact same thing with scripts here .
* If you go to the exercise files for this movie, 02\_02, you'll find an example .

Text

Description automatically generated

* Here, we have two files, index .
* html, and script .
* js .
* Script .
* js contains all our JavaScript, and in index .
* html, instead of writing the JavaScript in line, we instead used a script tag to reference the external file with the source attribute .
* The source attribute points directly at the file .
* The browser will encounter this line of code and say, "Oh, there's an external file the JavaScript .
* "Okay, let me grab that, render it out "and then do whatever the JavaScript says "before I continue rendering .
* " So let's test this in the browser to see what happens .
* Go to the exercise files for this movie, 02\_02 .
* Open it in the browser, and you get this nice white page with nothing on it .
* That's because there's an error here .
* So if I go to inspect my code, you'll see down in the console, it says Uncaught TypeError Cannot read property appendChild of null .
* All right .
* So if I go to my script here and look, I'll see down here at the very bottom, it says appendChild

Graphical user interface, text, application

Description automatically generated

* So that's the function the console is saying there's something wrong with, and if I look at what the function is appended to, I can see what's wrong .
* So it says grab the document .
* Then find the body element inside the document and then appendChild .
* The problem is in the HTML document, the script is run before the body element is created .
* So this is what I was talking about in the previous movie .
* The browser renders everything from the top to the bottom, so it'll render out the JavaScript before it gets to the actual content of the page .

Graphical user interface, text, application

Description automatically generated

* The error is telling me the script can't get access to the body element because the body element was never created .
* To fix this, I can then take the script tag entirely and paste it in below the body element .

Graphical user interface, text, application

Description automatically generated

* Save index .
* html, jump back in the browser .
* It's already been reloaded by the live server, and now we can see everything is working .

Graphical user interface, text, application, email

Description automatically generated

* There are no errors in the console, and the content is displaying as it should .
* So the JavaScript in the external file is interacting with the real document, and everything is loading in the order it's supposed to .
* Give feedback