**Troubleshooting JavaScript in the browser**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Instructor] Writing JavaScript is one thing. Getting it to work properly is quite another. As you start writing your own JavaScript, you'll often run into issues you need to troubleshoot. You've already seen the most heavily used troubleshooting tool in action throughout the course. It's the browser console. Here you can log anything from your code into the console to see what's going on. And you typically do this if you're capturing values and then want to see what the values are before you start working with them. So here in this code example I am grabbing this variable called strapArray and I just want the console log it out to see what it is. I'm also a console logging out an element down here. And when I jump over to the console, you can see the output. We have a NodeList with some items in it and I can burrow into them. And I'm also getting the output I want. So this console log is used all the time to send values to the console so you can see as you're iterating through your code what is happening and whether you're getting the values you're looking for. The console log is also a live coding environment. So you can write whatever code you want here and it's effectively like writing JavaScript in a file except you're doing it live in the browser. So you can run code in the browser and test it right away, directly from the console. If there are errors in your code, let me see, I'll introduce one here and save it. The console will also try to tell you what's wrong. Typically it'll tell you some sort of error message like Uncaught SyntaxError and then say, "missing parentheses after argument list" and it'll give you the file the error occurs in, in this case script.js and the file line. So here we are on line 50. In most cases this error message will be fairly accurate and will tell you what's wrong. So in this case it'll tell me there's a missing end parentheses. And when I look here on line 50, there's a missing end parentheses. And combined with the tools in the code editor, you should be able to figure out what's wrong. However, in some cases, due to how JavaScript works, you may have an error inside a function, for example, but the error gets kicked where the function is called. So it's not always immediately obvious where the error is sitting and what that error is. And these console log error messages are not always the most helpful. For this we have more advanced troubleshooting tools in the browser. Now I could create an entire course just focusing on these tools. So for now, let me just give you a quick preview. If you open the exercise files for this movie in the browser, then I'll open the developer tools and go to Sources. Here You'll see all the files that are currently being used in the browser. So if you're ever wondering, "I wonder if there's an image file or something else that's available" you can actually get a full breakdown of everything that's being run in the browser right here. What we're looking for is the script.js file. And when you click on this, you can see the entire script.js file here. So this is the same file that we created in the exercise files, and it's now being previewed in the browser. From inside this tool, I can now start troubleshooting by doing things like adding a break point, so that I can step through the code and see exactly what's happening. What I want to do here is grab this newStrapLength function. I want to see what is happening as this function is executed. So I'll take my mouse and click on line 43 here. This adds a break point. Now I need to reload the browser would this window open. Now at the very top of the page, it says, "Paused in debugger." So this is the debugger and I've currently paused the code here on line 43. Now what you're noticing is inside the debugger, I now get the value of strapArray displayed here next to it, so I can scroll through and look at everything that's being sent here. And as I stepped through this, and I can do that using this button over here or hitting F9 on my keyboard, I literally step through the code line by line and the browser will stop at each line and tell me what is going on. So here I can watch this listElement be broken into a forEach loop so that we now have a single listElement. And if I step to the next one, it says, this let side now has the value left and it comes from the listElement. And I keep stepping down. Now we have this const lengthform and here's the value of lengthform. And we can keep stepping down through this code and see exactly what's going on. Oh, this is really useful if your code isn't doing what you intend to do and you wonder what on earth is going on here. Are the wrong values being passed. Where is it crashing. Where's the values breaking. 'Cause you can see live in the browser what values are being passed down, when functions are kicking in and what they're doing. And you can add as many of these break points as you want by just clicking these lines here. And these break points will be triggered every time you reload the browser as long as you are inside the debugger. Like I said, these tools are well outside the scope of this course but now that you know they're there, I urge you to inspect your own code here and make yourself familiar with how they work. It is quite literally what I showed you. Just open up the code file, click on a line, and then step through the code to see what's happening.