Chrome DevTools

Assignment 9.4 - Client-Side Debugging

Devan K. Wong

July 21, 2020

Web 231

Google Chrome to a normal person is just a place where people can go and surf the web. However, to a developer Chrome allows for much more than just viewing an HTML page. Chrome offers a tool called Chrome Devtools which essentially helps an engineer break down their code and helps debug it. The great thing about it is that it is already built into Chrome so if you have the browser, have you DevTools.

Being an email developer, one thing that I use consistently is the elements panel. This panel allows you to check your rendered HTML and manipulate the DOM all on the browser. Sometimes when using HTML and CSS the output isn’t coming out exactly how you want it. DevTool allows you to hover over any part of your browser and shows you exactly CSS is being used in that element. For example, your headline looks off and you inspect it. Hovering over the headline you find that you have unwanted padding in there. Then you realize that instead of adding padding to all of your paragraph tags, you did it to all of your headlines instead, a common mistake. This technique is so simple but so effective to pinpoint those rendering issues. However, making adjustments to the DOM does not change the actual code. Once the issue is figured out, the original code must be adjusted.

Another great use of the DevTools is the console panel. This can be used in many different ways. One being that it can run Javascript in the command line. This is a great feature for beginner coders. They can run algorithms in the command line creating an interactive experience allowing them to understand how things work like for loops and booleans. It also teaches the order in which Javascript runs. Every developer has experience using “console.log” in their code. It validates that the functions are working and/or if the variables have values in them. This console not only shows the output but also pinpoints the errors that the code has such as an undefined variable or missing syntax. This is super helpful to help debug your code.

Another way to debug your code is using the Sources panel. The main feature is to be able to run snippets of the code and use breakpoints to help debug your code. Breakpoints lets you pause your code in the middle of its execution, and examine all values at that moment in time (Chrome DevTools, n.d). This sounds similar to the console.log method that is in the console panel. However, that requires manual input and using breakpoints avoids touching your original code and allows you to decipher the code right in the panel. You can add breakpoints to places like event handlers or a specific line of code.

Using Google’s Chrome DevTools allows you to manipulate and debug your original code without touching it. However, using this tool is a temporary fix. Eventually you will need to update your code so that your outputs will look clean and there will be no errors.

Citations:

1. Chrome DevTools. (n.d.). Google. https://developers.google.com/web/tools/chrome-devtools
2. Nwamba, Christian. “The Chrome DevTools Sources Panel.” Telerik Blogs, Telerik, 10 Oct. 2019, www.telerik.com/blogs/the-chrome-devtools-sources-panel-overrides-filesystem-snippets-page.