Client-Side Debugging

William T. Thomason

Bellevue University WEB-231

Client-Side Debugging

Client-Side debugging tools are valuable assets that a developer can user to troubleshoot code and test how their code reacts is different environments such as mobile devices. I personally use the Chrome developer tool on a regular basis and couldn't imagine not having it apart of my project workflow. These tools have a wide array of features available by default right in the browser.

One of these features is that you can manipulate HTML and CSS in real time and view the results in the browser. This can be done by opening the website in the browser, hovering your courser over the element you wish to alter. Then right click and then click on "Inspect" to open the developer tool. This will automatically show the code related to the element you right clicked on in the developer tools. You can also use keyboard shortcut Ctrl+Shift+I to open the developer tools as well, however this won't take you to the specific element. Inside the developer tool you will find the "Elements" tab on the top bar of the tool. From this tab you will have the other tabs at your disposal, the one we will focus on it the "Styles" tab. There you can manipulate the CSS either by adding or removing selectors and or declinations. When doing this you can see these changes in real time inside the browser. This is very helpful when trouble shooting how an element may be affected on different screen sizes.

Another helpful feature of the Chrome developer tool is the console. The console can be used to view messages, log messages, debug, and to view info, warnings, errors and console logs. This can be extremely useful when working with JavaScript. The info, warnings and errors can be expanded in the console by clicking the arrow next to the associated icon. By doing this it will show important information regarding to that specific event such as what problem what found, the line number, file name and any codes associated with it. You can even use the console to run JavaScript code such as the console.log function. This can be called on to view variables

to ensure you are getting the expected information such as a variable or object. Another helpful function that can help you view data from a webpage in the console is the colsole.table function this will take the information gathered and display it in a easy to read format of a table, this is great for outputting arrays.

You can use Chrome Developer tools to test a websites performance by using the "Performance tab at the top of the DevTools panel. This tool can be can help you find what is slowing your program down so you can troubleshoot ways to fix it. This can be great for larger web application with a lot of content to help isolate where the issue is when it comes to load time and webpage performance. To run this test, you navigate to the Performance tab click record then refresh the page you would like to test. After the page has loaded you click stop and view results.

References

Chrome DevTools . Tools for Web Developers. Google Developers. (219, January 29).

Retrieved February 5, 2019, from https://developers.google.com/web/tools/chrome-

devtools/.

Basques, K. (2019, January 29). Get Started With Analyzing Runtime Performance |

Tools for Web Developers | Google Developers. Retrieved February 7, 2019, from

https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/

West, M. (2015, May 21). Mastering The Developer Tools Console. Retrieved February

7, 2019, from https://blog.teamtreehouse.com/mastering-developer-tools-console