Lea Trueworthy

Client Side Debugger Assignment 9.4

Google Chrome comes with a handy set of web development tools in the form of DevTools. A code editor that runs in your browser, made by Google. Its original purpose is for building Chrome Apps, but it can be used for general web development.

They provide developers deeper access into their web applications and the browser. You can do everything from testing your viewport on a mobile device to editing your HTML/CSS on the fly, and even measuring the performance of your site's individual assets (Jackson, 2017). Just like with editing HTML, you can also change CSS in Chrome DevTools and preview what the result will look like. This is probably one of the most common uses for this tool (Jackson, 2017). In the Elements panel there are 2 super useful buttons: The first lets you add a new CSS property, with any selector you want but pre-filling the currently selected element. The second one lets you trigger a state for the selected element, so you can see the styles applied when it's active, hovered, on focus. If you click on the name of the CSS file that you edited, the inspector will open into the Sources pane and from there you can save it with the live edits you applied. having access to the css file allows the user the freedom to use it for experimentation or for creating and maintaining a style guide for a web site.

Workspaces enable you to save a change that you make in Devtools to a local copy of the same file on your computer.This could be useful when you need some real content to test your website instead of the conventional Lorem Ipsum. To do that, just go to the Console panel and run the following command:
copy(document.getElementById('content').innerText)
This command assumes that the content is nested under the element with the ID of content. Depending on the web page, the ID may vary. So, before running this command, copy(), you need to figure out where the content (that you intend to copy) is being nested.
After running this command, the content will be immediately saved to the clipboard, so you can simply paste it into your document (Firdaus, 2018). It is a helpful feature and incase you get lost, Workspaces is usually able to map the optimized code back to your original source code.

Lastly, using breakpoints in Javascript can allow the user to find problems in the code or even allows an easier method to clean up useless code. A breakpoint is a section/line of our code where we want the execution to stop so that we can carefully inspect the execution (Ganga, 2016). You can set breakpoints in Chrome DevTools by clicking on the line number you want to break. There are multiple styles of breakpoints but the most well-known type of breakpoint is line-of-code. This is when you know the exact region of code that you need to investigate. But line-of-code breakpoints can be inefficient to set, especially if you don't know exactly where to look, or if you are working with a large codebase. You can save yourself time when debugging by knowing how and when to use the other types of breakpoints.

This is a very short list of tools that Devtool uses to make a developers job less complicated. Luckily, these are extremely powerful tools and many developers would be somewhat lost with it.

References

Firdaus, T. (2018, December 03). 5 Useful Chrome DevTools Tips for Developers. Retrieved from https://www.hongkiat.com/blog/chrome-web-developers-tips/

Ganga, C. (2016, June). Debugging JavaScript with Chrome DevTools Breakpoints. Retrieved from https://scotch.io/tutorials/debugging-javascript-with-chrome-devtools-breakpoints

Jackson, B. (2017, January). Chrome DevTools - 25 Tips and Tricks. Retrieved from https://www.keycdn.com/blog/chrome-devtools

Swain, S. (2016). Things you probably didn't know you could do with Chrome's Developer Console. [online] freeCodeCamp.org. Available at: https://medium.freecodecamp.org/10-tips-to-maximize-your-javascript-debugging-experience-b69a75859329

Chrome DevTools | Tools for Web Developers | Google Developers. (2019, January). Retrieved from https://developers.google.com/web/tools/chrome-devtools/