Assignment 9.4 – Jake Seever 02/08/2019

Client-Side Debugging Tools

The first feature of web browser developer tool that I will analyze is pretty print. Pretty print is a Google Chrome DevTools takes minified code and makes it more easily readable for web developers (Jackson, 2017). Minified code is where the code is compressed into a smaller format. Code is then placed back to back without spacing or line breaks and is very unreadable for the majority of human eyes and brains. While minified code will help produce smaller sized documents that are processed quicker and easier by web browsers it is very hard for developers to read and interpret what is going on in that document. Pretty print takes the minified code and makes it appear in normal format in the source tab of the Chrome DevTools. You can access pretty print by click on the two curly braces in the right hand side windows of the DevTools window after a document from the web page has been selected. By simply clicking those curly braces the code appears in a normal standard format and is easily interpretable.

The second web browser development feature I would like to elaborate on is device mode. Through Google Chrome’s DevTools you can preview how your site or web page will appear though devices of varying screen sizes (Jackson, 2017). To access this tool you simply need to press “Ctrl + Shift + M” and then you are able to choose the device that you would like to emulate. This can range from various phone screen sizes, tablets screen sizes, and computer screen sizes. This is especially important due to the fact that there are so many varying screen sizes with the large variety of devices available in today’s market. As a developer it important that you are aware of how your product will appear visually on as many user devices as possible. In 2017, 63% of web browsing was done by users through a mobile device (Enge, 2018). The trend is progressing further and further in the mobile device direction. With the majority of web browsing being done from a mobile device it is a must for developers to take the appearance on different screen sizes very seriously.

The third web browser developer I will be elaborating on is Firefox’s accessibility inspector. With over 57 million Americans having some form of disability it is important to make sure that you are developing with that in mind (Interactive Accessability, n.d.). Firefox’s accessibility inspector provides a means to access the page's accessibility tree, allowing you to check what's missing or otherwise needs attention to meet the needs of those with disabilities (MDN Web Docs, 2019). This tools evaluates everything from roles, actions, naming, and values. It also evaluates the structure of your DOM nodes. The accessibility inspector also your descriptions of elements, verifies that you have proper keyboard shortcuts in place, the child count of your elements, the states of your links, and the presence of a list of all the accessibility related attributes that would be important for a person with a disability to have present when using your site. This feature of Firefox is turned off by default because it slows performance but can be accessed by turning it on in the accessibility tab of the Firefox developer tools.

# References

Enge, E. (2018, April 27). *Mobile vs Desktop Usage in 2018: Mobile takes the lead*. Retrieved from Stone Temple: https://www.stonetemple.com/mobile-vs-desktop-usage-study/

Interactive Accessability. (n.d.). *Accessibility Statistics*. Retrieved from Interactive Accessability: http://www.interactiveaccessibility.com/accessibility-statistics

Jackson, B. (2017, January 17). *Chrome DevTools - 25 Tips and Tricks*. Retrieved from KeyCDN: https://www.keycdn.com/blog/chrome-devtools

MDN Web Docs. (2019, February 7). *Accessibility Inspector*. Retrieved from MDN Web Docs: https://developer.mozilla.org/en-US/docs/Tools/Accessibility\_inspector