Jonathan Disla

Assignment 9.4

Web 321

Prof. Krasso

Client-Side Debugging

In order to properly troubleshoot and clear bugs in a program, it is necessary to debug. Luckily developers have tools such as the Chrome debugger which assists in finding the errors causing an application to fail. The elements tab in the Chrome developer tools, displays the entire markup of a site and allows for temporary edits to be made. The console shows some errors and where they occurred. The debugger allows a developer to pause during any point in the program to inspect any errors that may be occurring.

A front end developer may spend most of their time within the elements tab of the Chrome dev tools. Especially when dealing with third party code built by someone else, it is necessary to understand the structure of the page. A developer can simply click on or hover over an element to know where it belongs to, its id or class, its styles and many other properties. Knowing what the id or class of an element is crucial to know how to target the element. The element that is targeted can have its styles overwritten in this tab as well. When overriding styles, a developer can see in real time the effects (if any) the style rule has on the element[Davies]. On another note you can also test screen sizes as well within the tool.

All proper software editing program has a console including operating systems. The Chrome developer tool offers very powerful functionality that allows to:

* Find errors and where they may be located
* Check values of known variables
* Test functions and their results
* Test values from within the editor by using console.log

While there are many more functions of the console, it is important to point out that native javascript functions can be used to edit what is on the page on the fly such as design mode[Erik].

The sources panel lets a developer debug more properly by going in depth within the source code. This tool also helps to transcribe code that was written in a Javascript framework. The panel contains three main sections, the file navigator pane, code editor pane and javascript debugging pane[Basques]. The file navigator pane facilitates the user in loading the desired file. New files can be created and saved within this pane as well. Once a file is chosen it is opened in the code editor pane. The code editor pane displays all of the javascript code needed to run the application. It is within the debugging pane and code editor pane where the developer can set breakpoints. However, in the debugging pane the breakpoints can be more precise for example by event, listeners, DOM, and global listeners. Once a breakpoint is established for inspection, the code can be ran through a series of steps that stop after each line of code. This process allows for more accurate visibility of functionality in the desired line of code. The values of each statement returned is shown and code can be debugged successfully.

There are much more functionality within the Chrome developer tools. I touched briefly on map and testing responsiveness with the device sizing tool. With so much assistance, Chrome as well as Firefox are the most developer friendly browsers around.

Reference List

Basques, Kayce. “Get Started with Debugging Javascript in Chrome Dev Tools,” Chrome Dev Tools, July 7 2020.

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

Davies, Joshua. “Using the Chrome Web Developer Tools, Part 1: The Elements Tab.”

Command Line Fanatic, Accessed December 15 2020.

<https://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art033>

Program With Erik. “Chrome Dev Tools 101 Tips and Tricks for Web Devs.” YouTube, 20 September 2019.

<https://www.youtube.com/watch?v=ucWpMZoB7Jg&ab_channel=ProgramWithErik>