Google Chrome DevTools is used to quickly create, test, and debug problems that will help web developers build a better website and improve performance. The three things I would like to discuss are inspecting and editing the DOM, using the browser console to debug JavaScript codes and mobile simulation. These three things are one of the standard tools used by web developers to help them overcome obstacles and improve the website.

Web developers can inspect the DOM by clicking the inspector icon on the top-left of the DevTool. When you click on an HTML element, the element will be highlighted with blue color and then alter the HTML tag, add HTML tags, and change the HTML element styles. Web developer finds the inspector tool useful because you can modify the web page and see the effect immediately on the web page. These changes are not permanent and do not affect the main source files. You can clear these changes by refreshing the webpage, and the changes will go away. When you click on an element with the inspector tool, you will see the style panel displayed on the right of the DevTool by default. The web developer will use the style panel to view and modify the style of an HTML element in the web browser. This feature allows web developers to test color combinations, fonts, and web page layouts. Another great feature from the style panel is the auto-complete feature that provides different CSS properties without Google searching CSS properties.

Chrome DevTool has a nice feature called Device mode. This feature allows web developers to simulate a mobile device and test how the web page looks and feels on a mobile device. You can also use responsive viewport mode that allows you to drag the side handles to resize the viewport to any width you need, and you can specify values in width and height. The only thing that the DevTool cannot simulate is the architecture of mobile CPUs, which is very different from the architecture from desktop or laptop CPUs.

Google Chrome has a built-in browser console that allows web developers to debug JavaScript code. One of the ways to debug problems is by inserting console.log() statements into the code and output the results. When you see the result, the web developer can investigate issues and correct the problems. The web developers can enter JavaScript code directly into the console and interact with the web page. Another way to debug the problem is by using breakpoints that allows you to pause your code in the middle of its execution, and then web developers will examine all the values during that moment. Using breakpoint works better than using the console.log() for debugging because you don’t need to manually add console.log() statements to the source code. When you pause the code with breakpoints, the DevTools will show you the values from all the variables, and then you can check if these variables are affecting your JavaScript code.

**Reference:**

* How to use the Chrome Inspector and Debugger. (n.d.). Retrieved July 25, 2020, from <https://courses.cs.washington.edu/courses/cse154/19su/resources/assets/debugging/chrome-inspector.html>
* Simulate Mobile Devices with Device Mode in Chrome DevTools. (n.d.). Retrieved July 25, 2020, from https://developers.google.com/web/tools/chrome-devtools/device-mode
* Console Overview  |  Chrome DevTools  |  Google Developers. (n.d.). Retrieved July 25, 2020, from https://developers.google.com/web/tools/chrome-devtools/console