Google Chrome offers a lot of different options for in browser debugging. It’s possible to do everything from inspecting HTML to make sure tags, classes, and ids are correct to viewing and editing JavaScript to viewing network requests and replies.

Using the “Elements” tab [1] in the inspector allows you to view the page’s HTML and CSS. This can be useful for debugging to see if an element exists (it may not have been rendered if the code wasn’t written correctly), or if it has the correct id or class. This can be a great place to start debugging because you can easily check to see what is being rendered. For example, you may have written a JavaScript function that alters CSS properties, but those changes aren’t being applied. Using the “Elements” tab you can look at the HTML and make sure that your function is targeting the correct element. It’s easy to write a function and forget to add a class or id to the element you want to manipulate or apply the wrong selector. Examining the HTML and CSS lets you see exactly what properties an HTML element has when it’s rendered, and updates them live as you click a button, scroll, or otherwise interact with the page.

The “Console” tab [2] has some useful debugging features as well. If you’re testing to see if a function call happens, it’s easy to add a console.log(“test”) statement to the function and then check the console to see if “test” shows up. If it does your function is executing. There may be other issues that are causing problems, but you can at least narrow down the area of code you need to troubleshoot. You can also use the console.log function to output data and make sure you’re getting what you intended to. This can be especially important if you’re manipulating large arrays or complicated objects. The console will also show you errors, including content that was blocked by the browser or was unreachable. If your code isn’t working it could be because the link you were using for a JavaScript library is broken, meaning anything using that library won’t function.

The “Network” tab [3] has a lot of useful debugging information as well. The network tab will show you every request and response a page makes, including the document itself (the HTML), plus CSS, JavaScript, JSON, images, and more. This will let you see what was requested, the status (whether it succeeded or failed), the type, where the request was initiated, the file size, and how long it took to load. If parts of your page aren’t loading you can look at the network tab to see if the request is being sent, and if it is what the status is. If a request isn’t made, it’s obvious why something didn’t load. A 2XX status indicates success. A 4XX status indicates a problem on the client end. A 5XX status indicates a problem on the server end. This can help determine where exactly the issue is located.

Sources:

1. Get Started With Viewing And Changing The DOM. (n.d.). Retrieved from <https://developers.google.com/web/tools/chrome-devtools/dom/>
2. Console Overview | Tools for Web Developers | Google Developers. (n.d.). Retrieved from <https://developers.google.com/web/tools/chrome-devtools/console>
3. Inspect Network Activity In Chrome DevTools | Tools for Web Developers. (n.d.). Retrieved from <https://developers.google.com/web/tools/chrome-devtools/network>