Debugging JavaScript Lab

Now that we have some basic JavaScript, let's take a few minutes to get comfortable with the JavaScript debugger.

Examining network traffic

1. Using your browser of choice, navigate to the web page you were working on before.
2. Open the developer tools and look at the network traffic tab/section. Refresh your main page and examine each request and its corresponding response.
3. Look at how each request is generated and the assets that are returned from the server. Note that you can see images, scripts, pages and stylesheets.

Using the JavaScript debugger

4. Navigate to your page.
5. Open the JavaScript debugger tab/section.
6. Set a breakpoint and walk through the JavaScript code running on this page.
7. In the debugger, hover over some variables that you set and examine their values.
8. Stop the debugger and edit your JavaScript code. Add a couple of console.log statements. Here's one to try:

```
console.log("The random number", random);
```

9. Rerun your page and look at the console. Expand the treeview that you'll see for random. Look at all the stuff that is in there.

Debugging errors

10. Have one partner take a two minute break. The other partner should edit the page and introduce an error that will cause JavaScript to fail or the things on your page to not display right.
11. Have the first partner try to debug the page and discover and fix the error.
12. Switch roles and do it again.

HTML and CSS debugging

13. If you get finished early, run your page and examine the elements. Note that you can see the CSS settings (usually to the right side).
14. Practice making some adjustments to the HTML and CSS live in the debugger so you can watch the changes while you are looking at the page. Play some what-if scenarios with the look and feel of your page.
15. Take the ones that you like and redo those changes permanently in your actual HTML and CSS source files on the site.

Bonus! Getting fluent with other browsers

You've done all this in one browser, but remember that they're all slightly different. It is valuable to see what they have in common and how they're different for the times when your web apps are behaving differently in different browsers.

16. Re-do all the above steps except the intentionally introduced errors in your second-favorite browser.