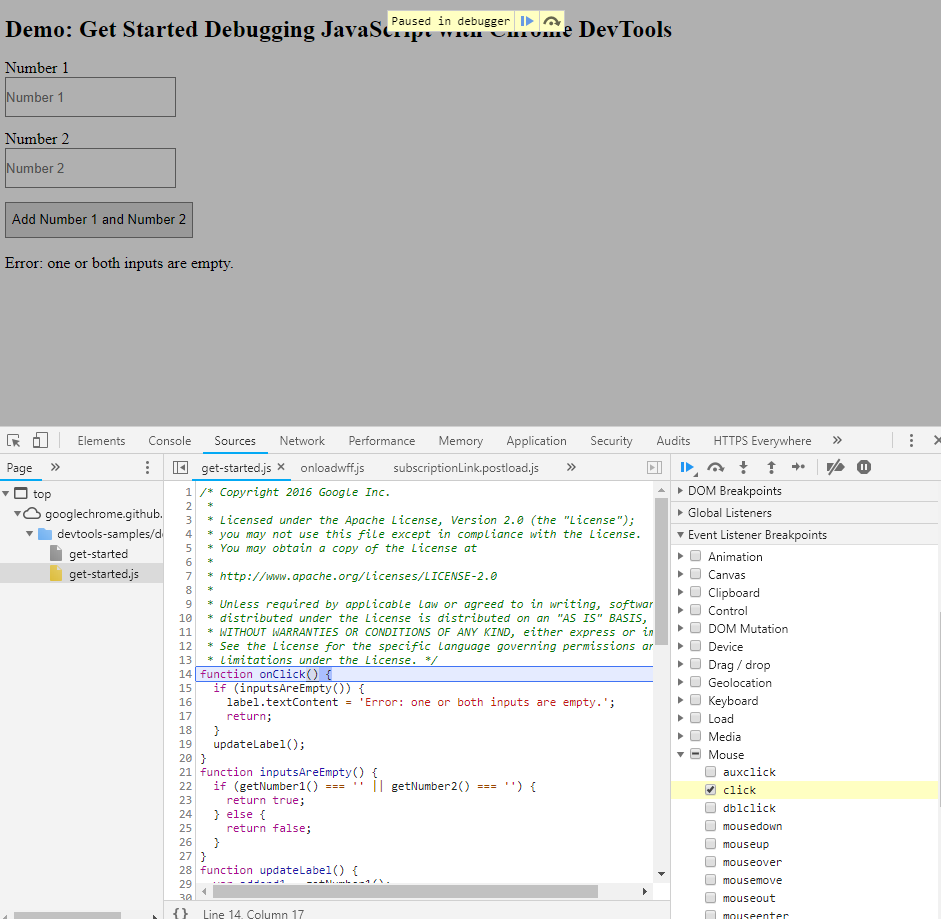
Debugging is a way of testing web coding. Any modern web browser can test code through the developer window, that is accessed by pressing F12 on the keyboard. The sources tab in the developer window is where you debug JavaScript. One way to test JavaScript is to set a breakpoint, which is a way to test a number of conditions. Setting a breakpoint on event listeners, such as a button click, or on a function, helps analyze potential issues within the code.

Figure 1 shows a test page that has two text inputs and a button, which simply adds the two numbers input in the text fields. Follow these steps to check the button functionality:

5.

4.

3.

2.

1.

Figure (Basques, 2018)

1. Once the page loads, press F12 on the keyboard
2. Add a mouse click event listener by expanding the Event Listener Breakpoint menu on the right-hand side of the debugger.
3. Click on the Add Number 1 and Number 2 button in the webpage.
4. The debugger will stop on the first line of code that shows the button, which is highlighted in blue in the center section of the debugger.
5. You can step through the rest of the code by pressing the play button located on the yellow toolbar at the top of the web document or the play button on the right-hand side of the debugger as pointed out by number one. Pressing this button will take you to the next line of code that relates to the button. You can step into the next function using the button highlighted by the number 3 in the image above and step out of the current function using the button highlighted by the number 4. This will allow you to skip the code in the current function and move onto the next function or get out of the current function. Once inside the code in a function, you can use the button highlighted by the number 5 above to go line-by-line within the function.

One can also select the HTML coding within the developer window and make temporary changes to the code. In the following example, the layout of the labels has been modified using the Elements section of the developer window. Figure two shows the original layout of the website. The following instructions will show you how to edit the code.

1. First, load the page in the browser and press the F12 key to access the developer window.
2. Select the Elements tab within the developer’s window as shown in figure 2.
3. Highlight the desired line of code in the elements section, shown by a gray highlighted line.
4. All of the applicable formatting for the highlighted option is shown on the right-hand side of the developer’s window.

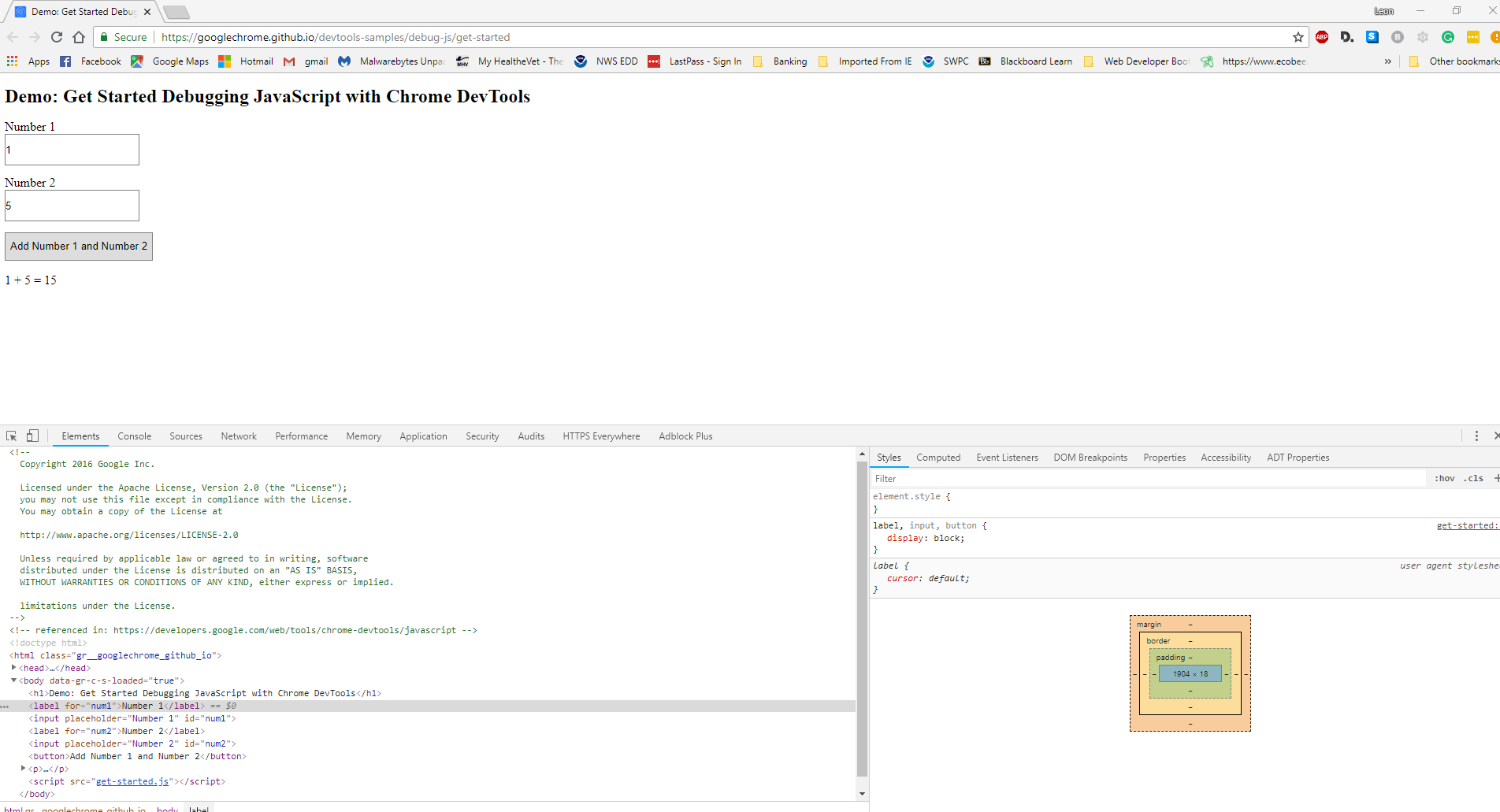


Figure (Basques, 2018)

Figure 2 shows the page in the original format. It is easy to make temporary changes to the site by using the styles pane. Figure 3 shows a few changes, one to the HTML code and another to the CSS styling.

To change the HTML:

1. The HTML was modified by changing the number 2 to 200000.
2. Hover the mouse over the line of code you want to change in the left-side of the Elements window. In this example, it was the label for the number 2.
3. Right click the mouse and select either the edit text option of the edit as HTML option.
4. Make your change and use the left mouse button to click outside of the box.
5. Your change is now reflected in the HTML

To change CSS styling:

1. In the elements window, left click on a line you want to format.
2. Click on the Styles option on the right-hand side of the developer window.
3. All of the options available to that item will be displayed.
4. In this case, the label for number 1 was selected in the elements window. The label, input, button style was changed by hovering the mouse over the display: block; CSS property, and a checkbox was displayed to the left of the property. Removing the checkmark from the box removed that CSS styling and caused the labels and boxes to display inline. Additionally, a color property was added to the text by adding two properties in the element.style for that label

Compare figures 2 and 3 to see the changes made to the display on the screen. These changes are not permanent and will be lost after closing the file.

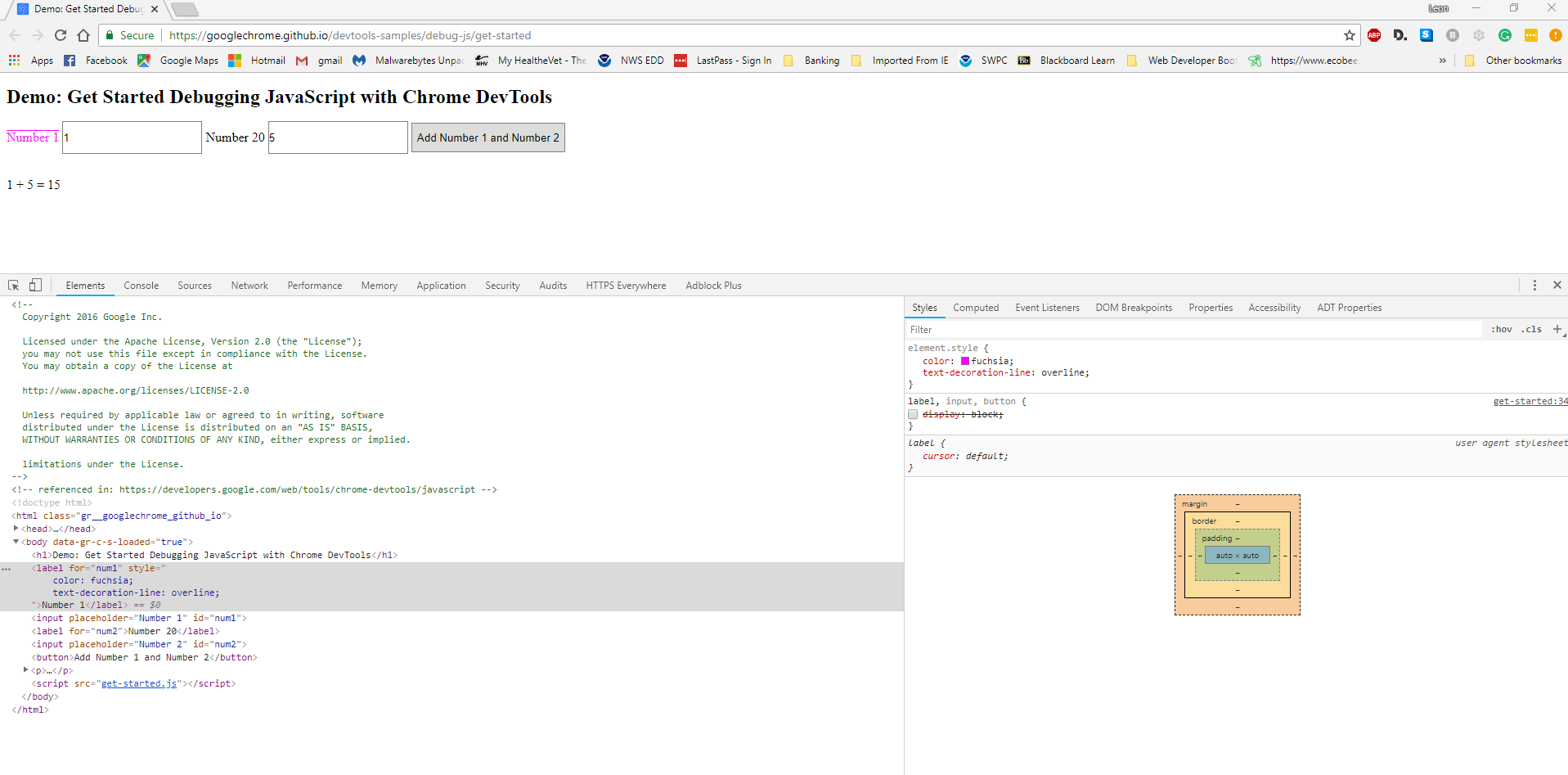


Figure (Basques, 2018)

Reference:

Basques, K. (2018). Demo: Get Started Debugging JavaScript with Chrome DevTools. Retrieved https://developers.google.com/web/tools/chrome-devtools/javascript/reference