LAB 6

**JAVASCRIPT 1: LANGUAGE FOUNDATIONS**

What You Will Learn

* Linking JavaScript into your HTML files
* The basics of JavaScript syntax

Approximate Time

The exercises in this lab should take approximately 90 minutes to complete. This is one of the longest and most important chapters in the book, so the lab for the chapter is also quite long.

**Fundamentals of Web Development, 3rd Ed**

Randy Connolly and Ricardo Hoar

Textbook by Pearson

Date Last Revised: February 7, 2022

|  |  |  |
| --- | --- | --- |
| **P R E P A R I N G D I R EC T O R IE S** | | |
|  | The starting In Class 6 folder has been provided for you (within the COSC-2328-01 -> Files -> In Class 6). | |
| 1 | If you haven’t done so already, create a folder in your personal drive for all the labs for this book. | |
| 2 | Copy the contents from In Class 6 from the folder to your course folder.  *Note: these labs use the convention of blue background text to indicate filenames or folder names and bold red for content to be typed in by the student.* | |
| 3 | At the top of each of the programs put the following comments:  <!-- In Class-6 – ex XX – COSC 2328 – Professor McCurry -->  <!-- Implemented by - Your name -->  Where ex XX is the exercise number | |
| 4 | Place comments in the body of your code!! | |

**Quick Tour of JavaScript**

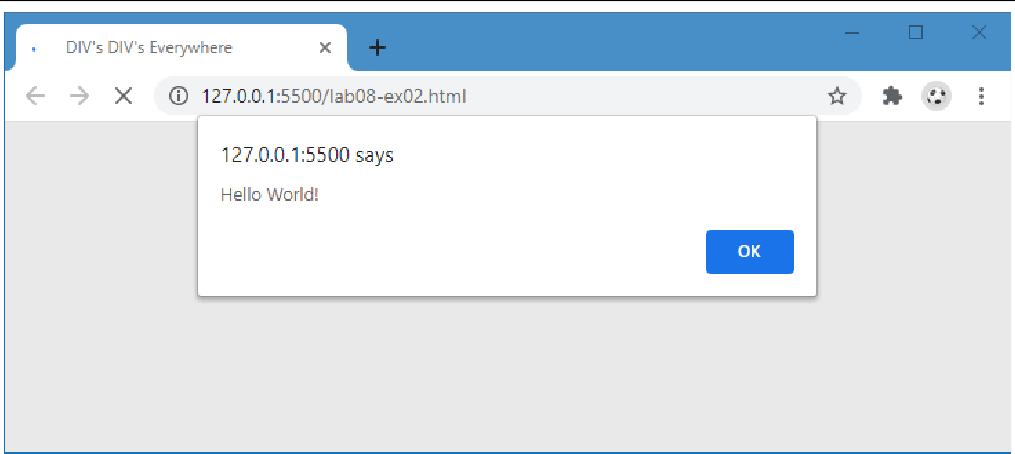
The next exercise shows you how to include JavaScript using the embedded technique.

|  |  |
| --- | --- |
|  | **E x er ci s e 6 . 2 — E M B ED D ED J A VA S C RI P T** |
| 1 | Examine lab06-ex02.html in your editor of choice. Some common editors include Visual Studio Code, Notepad++, Brackets, Sublime, and Eclipse. |
| 2 | Preview this file in your browser and it should resemble Figure 6.1. |



*Figure 6.1 – Beginning Exercise 6.2*

|  |  |
| --- | --- |
| 3 | Add the following to the lab06-ex02.html file inside of the <head> tags:  <script>  */\* Your first script \*/*  alert("Hello World!");  </script> |
| 4 | Save and test in browser.  *Notice that a popup is displayed with the text "Hello World!", and requires you to click ok before seeing the web page underneath. Depending on your browser and/or version, you will see something similar to Figure 6.2.* |



*Figure 6.2 – Finished Exercise 6.2*

|  |  |
| --- | --- |
|  | **EX ER C IS E 6 . 3 — E X T ER N A L J AV AS C R I P T** |
| 1 | Create a folder js in the folder that contains the lab06-ex02.html file.  Create a file named hello.js inside the js folder. Inside hello.js add the following line of JavaScript and then save the file:  document.write("this text was written from JS"); let count = 1;  document.write(" Count = " + count);  count++;  let output = "<br>Count = " + count; document.write(output);  *Depending on your editor and whether it is using a linter such as JSLint or ESLint, it is possible you may see warnings or error messages here about document not being defined. You can change that behavior by configuring the linter to use the browser as the environment; alternately, simply ignore those messages.* |
| 2 | Modify your HTML from Exercise 6.2 to include an external JavaScript file. Just after the  <body> tag, add the following link to the external file:  </head>  <body>  <script src="js/hello.js"></script>  <header class=centered>  <h1>Many DIVs to illustrate</h1>  </header>  *The above lines tells the browser to load a file in the same relative directory, named hello.js.* |

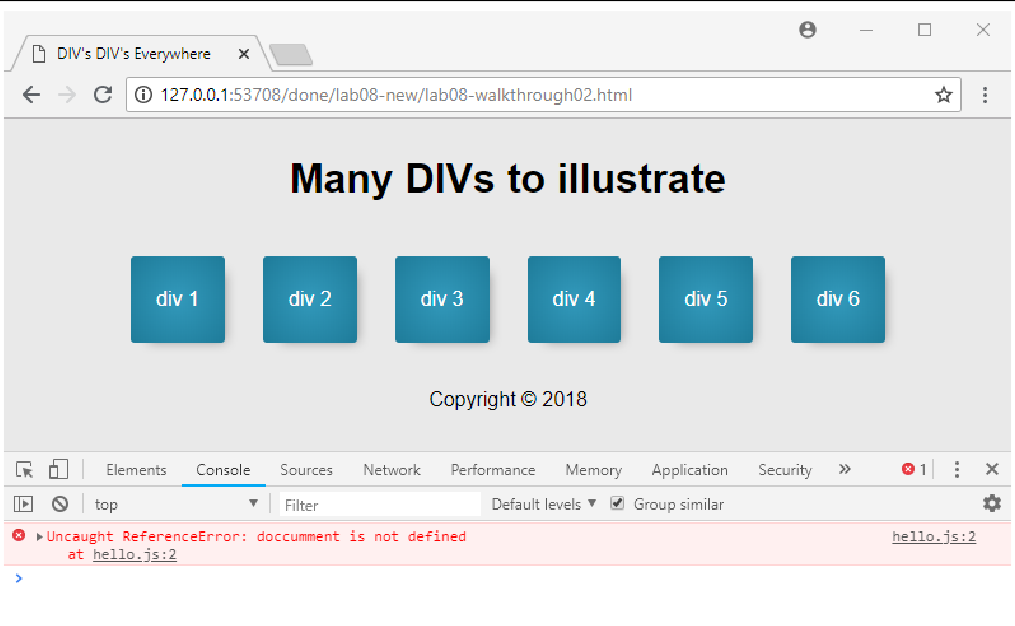


*Figure 6.3 – Finished Exercise 6.3*

|  |  |
| --- | --- |
|  | **EX ER C IS E 6 . 4 — U SI N G < N O S C RI P T >** |
| 1 | Although you have now used embedded and external JavaScript, you still may want to account for users without JavaScript enabled. Use the HTML file from the previous two examples and add the following markup after the <script> include from the previous exercise:  <script src="js/hello.js"></script>  <noscript>This page requires JavaScript be enabled</noscript> |
| 2 | Save changes and test in browser.  *In order to see if your tag is working correctly you must disable JavaScript.*  [*https://www.enablejavascript.io/en/how-to-enable-javascript-on-google-chrome#:~:text=Under%20%E2%80%9CPrivacy%20and%20Security%E2%80%9D%2C,%E2%80%9CAllowed%20(recommended)%E2%80%9D*](https://www.enablejavascript.io/en/how-to-enable-javascript-on-google-chrome#:~:text=Under%20%E2%80%9CPrivacy%20and%20Security%E2%80%9D%2C,%E2%80%9CAllowed%20(recommended)%E2%80%9D)*.*  *Once JavaScript is turned off, refresh the page and you should see the noscript message printed at the top of the page.*  Take a screenshot of your output and include it in your output submission.  Re-enable Javascript before moving to the next step. |

For the next exercise, you will build on the example from Exercise 6.4 so you can continue working in those files. Because you will eventually have multiple JavaScript files to manage you will now organize them into a folder like we did for our image and css files in previous labs.

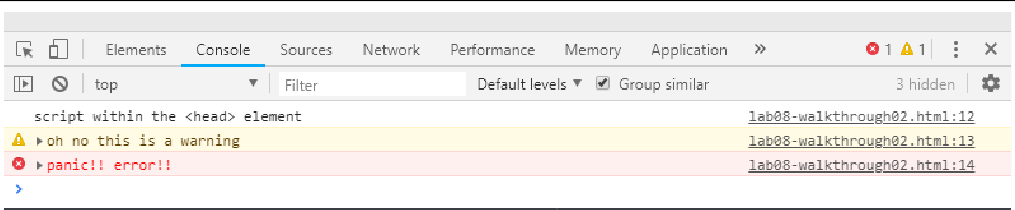
|  |  |
| --- | --- |
|  | **E x er ci s e 6 . 5 — U S IN G T H E B RO W S E R C O N SO L E** |
| 1 | Now you will purposely make a syntactic error in our JS file so you can learn to identify and fix errors. Make a copy of the first line that contains document.write in the js. Comment one of them out. Modify the code inside the hello.js file so that you misspell document as doccumment so that our line of code reads.  doccumment.write ("this text was written from JS");  *Now refresh the page and notice that the text at the top of the page is missing. However, despite the text missing there is no immediate notification that an error has occurred!* |
| 2 | To get better feedback about programming errors, you will want to use some type of JavaScript debugger/developer tool within your browser. |
|  | *If you are using Chrome, right click the webpage and click on inspect.*  *If you are using FireFox, use the Tools | Web Developer | Web Console menu option. If you are using Edge, then select the ellipse button, and choose Developer Tools. If you are using Safari, you will have to enable the Developer tools first within Preferences| Advanced and turn on the Show Develop menu check box. Once you do so, you can use the Develop | Show JavaScript Console menu option.* |



*Figure 6.6 – Chrome Console*

|  |  |
| --- | --- |
| 3 | Get a screen shot of your error and include this in your lab output submission. |
| 4 | Fix the error by spelling document correctly and then refreshing the browser. (comment out the bad line and uncomment the good line). |
| 5 | Return to the JavaScript console. You can use the console as well to run and test JavaScript. This can be especially useful way to interrogate the state of JavaScript variables. |
| 6 | Enter the following into the browser’s console: count  *After you press Enter, the console will display the current value of the count variable (which should be 2).* |
| 7 | Enter the following into the browser’s console: output |
| 8 | Enter the following into the browser’s console: var temp = count \* 8;  *You can also enter any valid line of JavaScript into the console as well. The console will display the message undefined. It displays undefined because the console evaluates every expression entered into the console: this assignment does not produce/return a value so it displays undefined.* |
| 9 | Enter the following into the browser’s console: temp  *This will display the current value of temp (which should be 16).*  *Get a copy of you console and save it to your lab Output file* |

|  |  |
| --- | --- |
| 10 | Switch to your source code editor, comment out the call to the alert() function, and add a call to the console.log() function as shown below. Test in browser.  <script type="text/javascript">  */\* Your 1st script \*/*  //alert("Hello World!");  console.log("script within the <head> element");  </script>  *The console.log() function outputs directly to the browser’s JavaScript console or provides a user message. This can be a helpful technique for debugging JavaScript. Several examples in this and future labs will make use of this (and the other) console functions.* |
| 11 | Add the following lines and test.  <script type="text/javascript">  */\* Your 1st script \*/*  //alert("Hello World!");  console.log("script within the <head> element"); console.warn("oh no this is a warning"); console.error("panic!! error!!");  </script>  *This will display messages in the console similar to those shown in Figure 8.6. You will learn more about debugging JavaScript in a future lab.* |
| 12 | *Get a copy of you console and save it to your lab Output file* |



*Figure 6.6 –Using the console object*

**Part B**

Fix any errors

Upload the lab06-ex02.html,js/hello.js to your Digital Ocean server and create a link to this webpage on your homepage (lab06-ex02.html). The webpage and images should display if I click on the link.

Submit all of your file(s) and images to Canvas and name the file:    
inclass6\_<yourSEUusername>.zip (the files should work on my computer if I unzip it)

Provide the link to the digital ocean homepage in the comments of your Canvas submission.