

**COSC 360**  
**Lab 5 - JavaScript**

**Information:**

This assignment has two parts. Part I and II will involve the submission of JavaScript and CSS files.

**Instructions:**

**Part I: Form Validation**

In this exercise you will use JavaScript to pre-validate a form before submission.

1. Download the provided HTML5 code and CSS. Place them in your local development directory. You will be able to view the page without a webserver. Examine the [lab5-1.html](#) file to understand how specific elements have been identified. Do not modify the HTML code.
2. Create an external JavaScript file called [lab5-1.js](#) and place it relative to your html file as [script/lab5-1.js](#). **Place all your JavaScript in this file.**
3. Create an external JavaScript file called [lab5-1-highlight.css](#) and place it relative to your html file as [css/lab5-1-highlight.css](#). Create a CSS style that can be used to highlight blank field (i.e. missing data). Have the style change the colour of the field to a colour that will draw the user's attention (red).
4. Setup a listener on the form's submit event so that the code prevents the submission of the form ([preventDefault\(\)](#)) if either the title of description field is left blank or the accept license box is not checked; otherwise submit the form.
5. Enhance the JavaScript so that blank fields trigger a change in the appearance of the form using the style from step 3.
6. Add another listener to the fields so that when the user types into a field (changed event), JavaScript removes the colour styling as the field is now valid.

**Testing:**

1. Test the form in your browser to make sure fields function correctly. Attempt to submit the form with either field blank and ensure the offending field is highlighted. Ensure that the page is not refreshed.
2. Type into the highlighted fields and check to ensure that the error colour is removed.

**Add your JavaScript and CSS file to your repository and push to GitHub. This will be validated against existing HTML.**

## Part II: Node Highlighting Helper

In this exercise you will write a helper script that could be used on any web page to help identify different elements simply by including the JavaScript file. Use your submitted code from lab 3 (the table and form lab) as a basis for testing. The goal here is use only JavaScript to create the desired end result as shown in Figure 1 below.

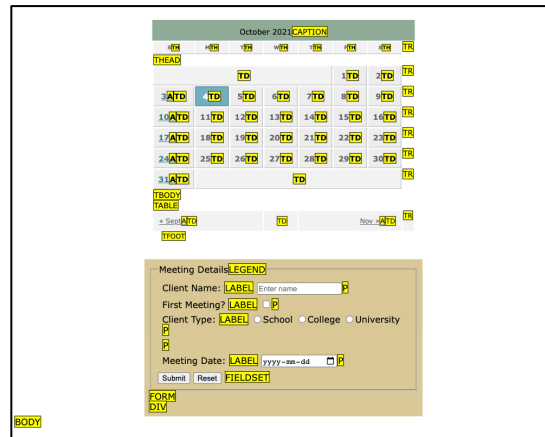


Figure 1 - Node Helper Example

1. Create a copy of your submission for lab 3 and rename it `lab5-2.html`.
2. Create a JavaScript file called `lab5-2.js` and place it relative to your html file as `script/lab5-2.js`. Place all you JavaScript in this file. Add the script to your HTML code so that it runs after the page is loaded.
3. Your script should navigate every element of the DOM for a given page. For each element in the body of the page, determine whether it is a `TextNode` (type 3) or not.
4. Have your script create a new child node for every **non-text** node encountered. The new node should take on the class `"hoverNode"` and `innerHTML` equal to the parent tag name. Define appropriate styles for this CSS class in your script so that the element is yellow (as shown in Figure 1).

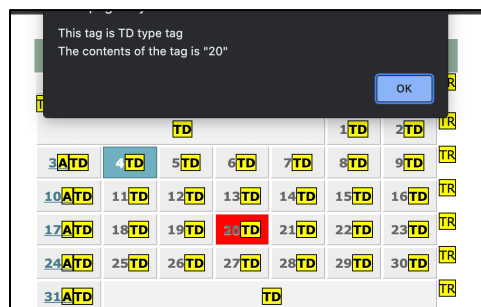


Figure 2- Alert Pop-up When Node is Clicked (in red).

5. Add listeners so that when you click on the newly created nodes, they will alert you to information about the tag name, so that when a node is clicked a pop-up alerts you to information about the node including its `ID` and `innerHTML` (Figure 2).

**Testing:**

1. Test your script by loading it onto any page (but start with the page from lab 3). All the tags should be identified, and yellow pop-up boxes are displayed. You can use your script on any other page just by copying the contents of your js file and pasting in into the JS console in your browser.
2. Reflect on how to improve/enhance this script into a more useful tool for web development and debugging.

**Add your JavaScript and CSS file to your repository and push to GitHub. This will be validated against existing HTML.**