**Project One Compare & Contrast (HTML only)**

**Opening sections:**

ExampleCode: They open with the <script> element. This is used to embed things like JS scripts.

Compare: We opened with a standard HTML opening that we picked up from freeCodeCamp (see below). They simply opened with a <div> element (“division” element). Based off what we learned from our tutorial, div is probably the most frequently used element in HTML. It divides up content, can contain elements that can be manipulated with CSS and JS, and any element can be nested inside of it. Their entire script is nested inside of this outer div element (div is used consistently within as well). They also nested their script inside the <form> element. This allows a webpage to take entered data and send it to a server sided source (ex: enter a name in text box, click enter -> data is sent to the server).

<!DOCTYPE html>

<html>

<head>

</head>

<body>

</body>

</html>

**General Observation:**

“id” global attribute used a lot across many different elements. Basic understanding thus far is that these uniquely id HTML elements to be used in styling’s/JS scripting.

Current confusion (possible Caleb question): Certain sections with related content have inconsistencies in the use of “id” and “for” attributes. Also, not understanding the usage of “for” in relation to related elements not having “id” of the same value as “for”. Current understanding is that “for” links elements together, IF the values are the same. I am not seeing that with their script.

**Section 1:**

I nested input elements inside paragraph elements for each option (5 total) that were all nested in a div element. They nested each option inside individual div elements and instead used label and input as individual elements (did not nest one inside the other).

**General Observation:**

Start using <label> instead of <p> for simple “labels”. This should be fairly intuitive to remember. A label’s a label is a label, right?

**Section 1 cont (Section 2: Checkboxes):**

ExampleCode: For radio buttons/check boxes, they used “label” followed by “paragraph” elements that nested their “inputs” which would be followed by the text that defines that input followed by a “break” element to go to the next line (making things vertical, not horizontal). This was definitely a more efficient use of space than our solution. Our radio buttons were divided up into div elements nested with “input” followed by “label” (Also used for the date input). Our checkboxes are probably the clumsiest part of this script. Each checkbox divided up into their own <div> with nested label elements which further nested the input elements. The example code used a more consistent form between their radio buttons/checkboxes: <div> -> <label> -> <p> -> <input>. We could have used the template for our radio buttons, but didn’t for some reason.

QUICK ASIDE: Relating to the last sentence above, we were utilizing the freeCodeCamp examples in parallel with our project scripting which may have contributed to us trying different ways of writing out the elements. NOTE: Be more consistent moving forward. Find better templates for rendering particular content that is more efficient.

**Section 3:**

Our code lined up precisely with the example code. Consider that a win?

**Section 4:**

ExampleCode: Text boxes followed the same general pattern. <div> -> <label> -> <textarea>

\*<p> used in first text area pattern, not in second.

Our code followed the general pattern of <p> -> <textarea>. <div> was used for the first button submission; the second was standalone (example nested the first in <div>, set a ”for” & “id” link in the opening tag).

**General consensus:**

ExampleCode:

* More consistent in terms of pattern structures for similar types of content.
* Used more attributes to extend the utility of their elements.
* More efficient use of elements which simplified their code more than ours. Factoring comments, their code was probably 50-60 less lines than ours; more elegant and easier to read too.

Take-away

* Work on understanding elements and their attributes more. This should allow you to figure out more efficient ways of structuring your scripts.
* Write out comments BEFORE you start writing your script. Doing this should also give you an idea of which types of elements you will need (USE THE DOCS).