COP 3813 – Introduction to Internet Computing – Fall 2015

Homework (HW) 4 – Unit Conversion Using JavaScript

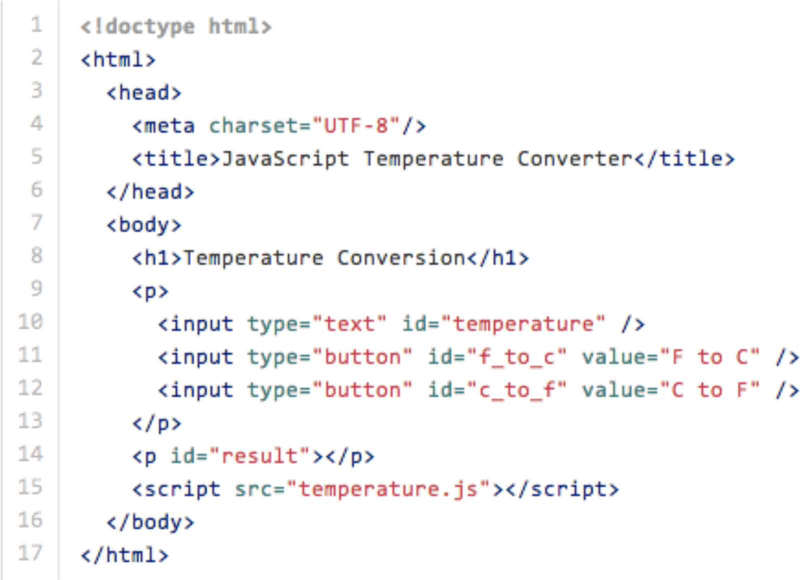
Goal

Design a simple, visually pleasant, JavaScript-based “app” that performs a useful function, namely, unit conversion.

Recommended Procedure:

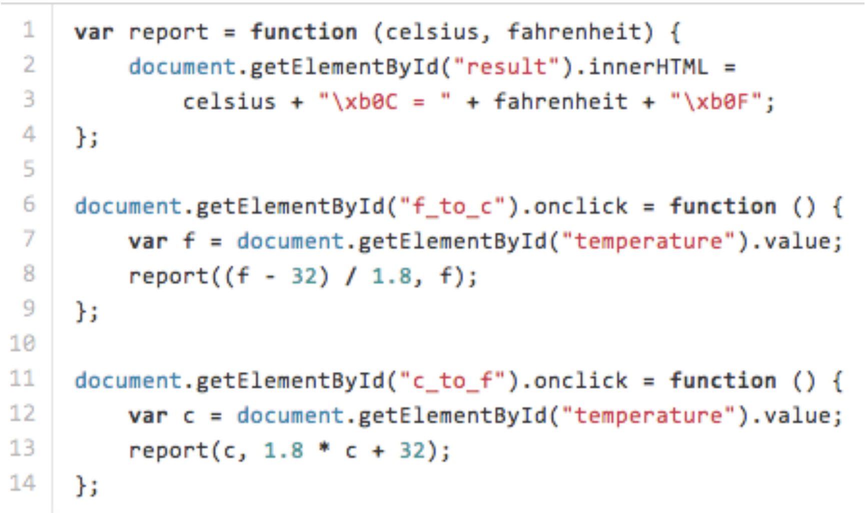
1. **Start from a working example**, such as the ‘temperature conversion’ code from https://github.com/dondi/javascript-book/tree/master/chapter02/temperature (also available on Bb, and reproduced below for your convenience).

temperature.html:



```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>JavaScript Temperature Converter</title>
  </head>
  <body>
    <h1>Temperature Conversion</h1>
    <p>
      <input type="text" id="temperature" />
      <input type="button" id="f_to_c" value="F to C" />
      <input type="button" id="c_to_f" value="C to F" />
    </p>
    <p id="result"></p>
    <script src="temperature.js"></script>
  </body>
</html>
```

temperature.js:



```
var report = function (celsius, fahrenheit) {
    document.getElementById("result").innerHTML =
        celsius + "\xb0C = " + fahrenheit + "\xb0F";
};

document.getElementById("f_to_c").onclick = function () {
    var f = document.getElementById("temperature").value;
    report((f - 32) / 1.8, f);
};

document.getElementById("c_to_f").onclick = function () {
    var c = document.getElementById("temperature").value;
    report(c, 1.8 * c + 32);
};
```

2. **Run the example and ensure that you understand it**.

3. **Modify the example in a meaningful way**, replacing the temperature conversion functionality with other types of conversion, such as: metric to US units for length, mass, etc.
4. Improve the visual aspects of the app (use a **Bootstrap** template).
5. **Test your app** after every significant change / addition.
6. Once you've reached a point where your app is complete and fully functional in the browser of your choice (Chrome, Firefox, Opera, or Safari), **prepare the final package** (single zip, all that is needed, and nothing else).
7. **Submit the final package** via Blackboard.

Minimum requirements:

- Your app **must** be your own work. If you use a site, textbook example or any other source as "inspiration" along the way, please make a note of it in your report.
- Your app should perform a (set of) meaningful task(s).
- It is OK to reuse text, images, etc. from previous HW assignments.
- Your page should demonstrate separation between presentation (CSS), content (HTML5), and interactive functionality (JavaScript).

Deliverables

- **A single zip file** containing all files (.html, .css, .jpg, .js, etc.) necessary to see your page in a browser window and your brief report (see below).
- A very brief **report** (1-3 pages) describing any relevant aspect that I cannot tell just by looking at the page files.
    - For example: the editor(s) you used, the browser(s) you tested your page on, how was the JavaScript learning curve, which tasks were more time consuming, etc.

Please name your file using your FAU / Blackboard username as a filename, e.g., jsmith85.zip.

- **Grading rubric:**

JavaScript code – correctness:	**20%**
JavaScript code – functionality:	**20%**
JavaScript code – quality of code (and inline comments)	**15%**
Overall presentation (alignment, colors, fonts, images, etc.):	**20%**
Separation between HTML, CSS, and JS:	**20%**
Available online at: http://lamp.cse.fau.edu/	**5%**

- **Submission**: Please **submit your files via Blackboard** using *Blackboard's assignment submission option*. **Please don't hand in any assignment by email.**

- **Deadline**: September 28, 2015 - 11:59 p.m. ET

- These guidelines may be updated. Please check your email and Blackboard often.