# Assignment Title – Online Store – Part 3 (10%)

This assignment relates to the following Course Learning Requirements:

CLR 1: Develop content for websites and web apps using modern standard semantic HTML5 elements.

CLR 2: Design components and layouts using the latest version of Cascading Style Sheets.

CLR 3: Illustrate good design practices for typography, colour, layout, and navigation that adhere to the latest design standards and best practices for different platforms.

CLR 4: Enlist the use of modern web tools, IDEs, frameworks, and libraries to efficiently create websites.

Objective of this Assignment:

Modify the existing design of your online store from Assignments 1 & 2 to be responsive across platforms, and begin adding JavaScript functionality.

# Pre-Assignment Instructions:

1. To prepare you for this assignment, read the content of modules 1 – 8, follow the embedded learning activities, complete the module self-assessments, and complete Assignments 1 & 2.
2. Create a project folder on your workstation named CST8117\_Assignment\_3\_*lastname*\_*firstname*. Replace the values *lastname* and *firstname* with your own last name and first name.

Example: CST8117\_Assignment\_3\_Doe\_John

1. Download the resource assignment\_file\_headers.zip. This file contains the templates for file headers you will add to your assignment.
   1. **Right-click** on the assignment\_file\_headers.zip file, and select **Extract All…**
   2. Click the **Browse…** button, and navigate to your project folder, then click the **Select Folder** button.
2. Click the **Extract** button to unzip the header template files into your project folder.

**Assignment Tasks:**

1. ~~Copy the files from your Assignment 2 folder into your Assignment 3 folder.~~
2. ~~In the assets folder of your project, create a new JavaScript file named~~ **~~main.js~~**~~.~~
3. ~~From the project folder, open the file javascript\_file\_header.js in a plain text editor.~~
   * **~~Copy~~** ~~the JavaScript file header template from javascript\_file\_header.js.~~
   * **~~Paste~~** ~~the file header into the top of main.js, and complete the fields in the header.~~
4. ~~In the~~ **~~main.js~~** ~~file write a function named~~ **~~isValidEmail~~** ~~that takes a single String parameter named~~ **~~email~~**~~.~~
   * ~~This function should~~ **~~return~~** ~~the Boolean value~~ **~~true~~** ~~or~~ **~~false~~** ~~depending on if the email address received as a parameter is a valid email address.~~
     1. ~~There are a few ways you could test the email address, the following of which are acceptable:~~
        1. ~~Use what is called a regular expression to test the email address for validity.~~
           1. ~~Hint: Google “JavaScript regular expression email”~~
           2. ~~CITE YOUR SOURCES!!!~~
        2. ~~OR – use JavaScript String functions to check that all of the following conditions are true:~~
           1. ~~The email address contains an ‘@’ symbol, that’s position is at least the 3~~~~rd~~ ~~character in the string, and is followed by at least 5 characters.~~
           2. ~~The ‘@’ symbol is followed by at least one “.” character (for a top-level domain, such as .com or .ca).~~
           3. ~~The final ‘.’ In the string is at least 2 characters before the end of the string (to allow for at least a 2-character top-level domain).~~
           4. ~~Aside from the ‘@’ symbol, the only other characters are: dots ( . ), underscores ( \_ ), hyphens ( - ), and alphanumeric characters (A-Z, a-z, 0-9). No other special characters are allowed in an email address.~~
           5. ~~The first character must be alphabetic (not numeric).~~
   * ~~If the string is not a valid email address, or the parameter is not a String at all, return~~ **~~false~~**~~.~~
   * ~~Note that this function will be used in Assignment 4.~~
   * ~~Add the following test statements to your main.js file,~~ **~~below~~** ~~the isValidEmail function, to test the function:~~
     1. ~~console.log(isValidEmail("myEmail1@email.com"); // true~~
     2. ~~console.log(isValidEmail("my.Email1@e-mail.com"); // true~~
     3. ~~console.log(isValidEmail("myEmail1@email.c"); // false~~
     4. ~~console.log(isValidEmail("m@email1.com"); // false~~
     5. ~~console.log(isValidEmail(1); // false~~
     6. ~~console.log(isValidEmail(true); // false~~
     7. ~~console.log(isValidEmail("myEmail1@com"); // false~~
     8. ~~console.log(isValidEmail("my\_Email1@e.com"); // true~~
     9. ~~console.log(isValidEmail("1m@email.com"); // false~~
     10. ~~console.log(isValidEmail("my!Email@email.com"); // false~~
5. ~~In the~~ **~~main.js~~** ~~file, write a function named~~ **~~isAgeOfMajority~~** ~~that takes three number parameters named~~ **~~birthYear~~**~~,~~ **~~birthMonth~~**~~,~~ **~~birthDay~~**~~.~~
   * ~~Validate the parameters:~~
     1. ~~If~~ **~~birthYear~~** ~~is not a number or birthYear is not between 1920 and 2010, return false.~~
     2. ~~If~~ **~~birthMonth~~** ~~is not a number or birthMonth is not between 1 and 12, return false.~~
     3. ~~If~~ **~~birthDay~~** ~~is not a number or birthDay is not between 1 and 31, return false.~~
   * ~~Create two JavaScript date objects:~~ 
     1. **~~TODAY~~** ~~– a~~ **~~const~~** ~~Date object representing the current date~~
     2. **~~birthdate~~** ~~– a~~ **~~var~~** ~~Date object, created using the Date() object constructor. Concatenate the parameters to a string in the format “M/D/YYYY” before passing them to the Date constructor.~~
        1. ~~What are the possible return values from the Date constructor?~~
        2. ~~Why concatenate them into a string, instead of simply passing each number to the constructor as an individual date part?~~
        3. **~~Answer these questions~~** ~~in a comment in your code, inside of your~~ **~~isAgeOfMajority~~** ~~function.~~
     3. ~~If the difference between TODAY and birthdate is greater than 18 years (or greater than 6570 days – approximately), return~~ **~~true~~**~~.~~
     4. ~~Otherwise, return~~ **~~false~~**~~.~~
     5. ~~Hint: Google “date difference JavaScript”.~~
     6. ~~CITE YOUR SOURCES~~
     7. ~~Note that this function will be used in Assignment 4.~~
     8. ~~Add the following test statements to your main.js file,~~ **~~below~~** ~~the isAgeOfMajority function, to te>st the function:~~
        1. ~~console.log(isAgeOfMajority(2005,2,25); // true – just old enough~~
        2. ~~console.log(isAgeOfMajority(1997,2,29); // false – not a leap year~~
        3. ~~console.log(isAgeOfMajority(2008,5,1); // false~~
        4. ~~console.log(isAgeOfMajority(2000,1,1); // true~~
        5. ~~console.log(isAgeOfMajority(1980,12,31); // true~~
        6. ~~console.log(isAgeOfMajority("1980","12",31); // false – wrong data type in parameter~~
6. Add **CSS media queries** to your **main.css** file (and possibly **products.css** file, depending on your design).
   * Include media queries to test for portrait or landscape **orientation**, and **mobile** and **desktop** designs. You do not need to consider tablets at this time for this assignment.
   * **Add comments** to your media queries to indicate your understanding of their purpose and usage.
   * For mobile displays, switch to using a **hamburger menu**, and consider moving your main navigation menu to the bottom of the page.
     1. W3Schools Mobile Navigation Bar: <https://www.w3schools.com/howto/howto_js_mobile_navbar.asp>
     2. fullPage.js Blog: <https://alvarotrigo.com/blog/hamburger-menu-css/>
   * For mobile displays, reduce the number of columns of product images that are displayed. For example, on the desktop you might display 3 or 4 products in each row, whereas only 2 on each row on a mobile device.
   * For mobile displays, add 1-2 **Call to Action button(s)** to direct the user to one (or two) locations on your site. This could mean adding a large Products button, a large Contact Us button, a large Search option, etc.
7. Ensure that all fonts on your website are **relative** (em or rem), and are **at least 16px** in size.
8. Ensure that all images on your website use **relative** sizing, or have a **max-width** set.
9. ~~Add the~~ **~~viewport~~** ~~meta tag to the head section of every page on your website, with the~~ **~~device-width~~** ~~value and~~ **~~initial scaling~~** ~~set to 1.~~
10. Create a separate reference page that lists all of your sources that you have cited in text. Sources include software used, and images. Also, personal communications from a professional in the field count as a reference source.
    * For citation and referencing examples, see <https://owl.purdue.edu/owl/research_and_citation/apa_style/apa_style_introduction.html>.
    * Source code snippets used in assignments are cited using comments within the code. For citation examples, see: <https://uark.libguides.com/CSCE/CitingCode>
11. Your assignment is to be submitted in **ZIP** format (not GZIP, RAR, or any other compression method). It should be submitted with the following guidelines:
    * Ensure that your project folder is named CST8117\_Assignment\_3\_*lastname*\_*firstname* (with your first name and last name in the folder name).
    * Ensure the every HTML file, assets/main.css, assets/products.css, and assets/main.js files have complete file headers at the top of the document.
    * Navigate to the directory containing your project folder. **Right-click** on the project folder, and select **Send To > Compressed (zipped) folder**.
    * The resulting file name of the new ZIP file should be CST8117\_Assignment\_3\_*lastname*\_*firstname*.zip.

**Assignment Grading Rubric (10%)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Excellent**  **3** | **Good**  **2** | **Requires Improvement**  **1** | **No Attempt**  **0** | **Points** |
| **Assignment Quality** | Web site executes and displays as expected.  All required elements are present on the site.  The console displays no errors. | Web site executes and displays mostly as expected.  Majority of required elements are present on the site.  The console displays no errors. | Web site partially loads, and displays although not as expected.  Some of the required elements are present on the site.  Any warnings or errors do not stop the page from loading. | No effort visible in complete the work as assigned.  Few to no required elements are present on the site.  Errors prevent the page from loading. | /3 |
| **Assignment Knowledge and Skills Demonstration** | Clear and optimal usage of the HTML, CSS, and/or JavaScript technologies to design a website, present information, and solve problems. | Evidence of understanding of HTML, CSS, and/or JavaScript technologies to attempt to design a website, present information, and solve problems. | Weak application of HTML, CSS, and JavaScript technologies to design a website, present information, and solve problems. | No attempt made to use HTML, CSS and/or JavaScript to design a website, present information, and solve the given problems. | /3 |
| **Assignment Structure** | Assignment submission follows naming and formatting instructions.  Code is indented properly and consistently throughout the assignment (tab, 2- or 4- spaces).  Code contains descriptive comments from the student.  Best practices and naming conventions are followed. | Assignment submission closely follows naming and formatting instructions.  Code is mostly indented and fairly consistent throughout the assignment (tab, 2- or 4- spaces).  Code contains some descriptive comments from the student.  Best practices and naming conventions are attempted. | Assignment submission naming and formatting requires improvement to meet instructions.  Code indentation does not conform to standard levels and/or is inconsistent throughout the assignment.  Code contains little to no comments from the student.  Small attempt at following best practices and naming conventions is apparent. | Assignments submission does not follow naming and/or formatting instructions.  Code is not indented properly or consistently.  Code contains no descriptive comments from the student.  Little to no attempt to follow best practices and naming conventions. | /3 |
| **Citations** | No issues with using APA citations, references, or in-code citations. | Minor issues with using APA citations, references, or in-code citations. | Incorrect use of APA citations, references, or in-code citations. | Missing APA citations, references, and in-code citations. [[1]](#footnote-1) | /3 |
| **Total Points:** |  |  |  |  | /12 |

1. If there is evidence of missing citations or references, this could constitute plagiarism and will receive a grade of ‘0’. The regulations of academic dishonesty will apply. [↑](#footnote-ref-1)