COMP 10259: Assignment 2

Introduction & Overview

In this assignment you will practice using Bootstrap's JavaScript components and writing solutions to problems using JavaScript from scratch.

The problems are intended to refresh your memory on problem solving with computers, like Programming Fundamentals, before we move into more complicated, multi-component solutions. Though you might find solutions to these problems on the Internet, you are reminded that the only value these exercises have is as practice to you. Googling answers != research, thinking = learning. Any detected incidents of academic integrity violations will be cited.

The goals of this assignment are:

1. Practice modifying Bootstrap component sample code.
2. Practice writing software solutions to problems.
3. Practice reading input from a web page using JavaScript.
4. Practice writing output to specified HTML elements using JavaScript.

All elements in your submission must be styled to some degree but it is not expected that you try to create a work of art. A neat, professional design will be very sufficient.

If something is not specified, then you have flexibility to be creative. If you are uncertain, please post a question in the discussions area.

Details

Files and layout

Create 1 HTML file, 1 JavaScript file, and at most 1 CSS file. Name your HTML file index.html. You may pick appropriate names for your JavaScript and CSS files though you are encouraged to use all lowercase letters, and no spaces. You are encouraged to use a logical directory structure for your CSS, images, and other resource files. All CSS must appear in external style sheets. All JavaScript must appear in your external JavaScript file and ***must*** be loaded in your **head** element.

Content

You are expected to use your own creativity to design this web application and talents to implement it. As such, there is no standard solution to show.

Bootstrap

From the list of components on the Bootstrap 5 website[1], choose a component, other than the Carousel, to present the output from your solutions to the problems described below. Some suggestions of appropriate options include "Accordion", "Collapse", "Modal", "PopOver", "Scrollspy", and "Toasts". You are permitted to use other options, the only restriction is to "Carousel" because it was used in the class example. You are allowed and expected to copy the code from the the Bootstrap website as a starting point. You will modify it to accomplish the task below.

[1] https://getbootstrap.com/docs/5.0/customize/components/

The content of each section of your chosen component (similar to each slide in the Carousel) will be the output to a programming problem that you have solved in JavaScript. Each of these containers should have an **input** element and a **textarea**. The **textarea** must have the attribute **readonly** and will be used for your solution output. You should give them **id** attributes to make selecting them with JavaScript simpler. Simple patterns like **id="q1_input"** and **id="q1_output"** will make it easy to label all of the elements uniquely. Each input element should have an **onchange** attribute that corresponds to the function you create (see below for a naming suggestion). **Note**: do not put your inputs inside a form, this way you won't have to prevent the submission of the form.

JavaScript Problems

From each of the groups below pick the specified number of problems to solve in JavaScript. Solve each problem in its own function. Name your function group_x_choice_y based on the labels below. You may create additional helper functions as needed. It is expected that you may need to look up syntax on external sites like w3schools, but it is strongly recommended that you work out your pseudocode solution first for the practice with critical thinking.

Group 1 – Select 2

1. Read the input from the document, if the input is numeric and if the input is 0, or if it's between 13 and 17 inclusive, output "In range" otherwise output "Out of range". If it's not a number output "Not a number".

2. Read the input from the document and if it is a positive number interpret it as some number of days. Convert it into years, months, days. To make things easier, assume there is 365.25 days per year and that every month is 30.44 days. Left over days output may have a decimal portion, but years and months should be integers. Output these values with appropriate labels. If the input is not a number output "Non-numeric or negative input".

   Note: decimal math is prone to slight rounding errors, don't be concerned about an answer like 14.779999999 vs 14.78 if you see it. For added challenge use Math.round to guarantee 2 decimal places.

   Example: input: 2175.87 output: 5 years, 11 months, and 14.78 days

3. Read the input from the document and interpret it as the length of the side of a square if it is numeric and greater than or equal to 0. Output the perimeter (4*side) and area (side*side) of the square. If it is not a number output "Can't calculate".

Group 2 – Select 2

For all questions in this group use only loops and substring operators, absolutely no regular expressions, or use of reduce/replace/replaceAll/include/etc methods.

1. Read the input from the document and if the input is "a", "e", "i", "o", or "u" output "A Vowel". If the input is "y" output "Sometimes". In all other cases output "Not a vowel". Your function should be case insensitive, i.e., "A" or "a" would both show "A Vowel".

2. Read the input from the document. If it is an integer, split the absolute value of the integer apart and add up all the digits. If the resulting sum is greater than 9, repeat this process until the result is a single digit. Output the result of the calculation. If the original input was not an integer output “Bad input.”

3. Read the input from the document. If it is a positive integer, output the factorial of the input value. Otherwise, output “Cannot compute factorial value”. The factorial function is defined as multiply all the numbers from 1 to the value together. For example, the factorial of 5 is 5*4*3*2*1 = 120. Only compute this value using a loop, no other formulations are allowed.

Group 3 – Select 1

1. Read the input from the document. This string must be between 10 and 20 characters long, inclusive, and consist of only 0’s and 1’s. If the string does not meet these conditions, output the result 0. If the input is valid then interpret it as a binary number. Using only loops and substring operations calculate the integer value of the binary number. For simplicity, we will assume “little endian” numbers … the smallest number is on the left and get bigger moving to the right. Recall that binary values are powers of 2, with the first few values being $2^0 = 1$, $2^1 = 2$, $2^0 + 2^1 = 3$, etc.
2. Read the input from the document and treat the input as a short equation with one operator. The operators you should expect are “+”, “-“, “*”, “/”, and “%” with their usual meanings. All valid formulae will have a positive number, an operator, and a positive number, i.e., “3 * 2” or “5 + 2”. Ignore spaces. If the formula is valid, output the result of the formula, otherwise output “invalid formula”. Use the following inputs as test values “*2”, “3*”, “3+*2”, “3*2”, “15*10” which should result in outputs “invalid formula”, “invalid formula”, “invalid formula”, “6”, and “150”.

Misc. Requirements

- You must use Bootstrap 5 to achieve the major styling of your page. Any other version will result in a grade of 0. You are allowed to augment your BS CSS with your own custom elements, but it shouldn’t be most of the styling.
- Use semantic elements wherever possible.
- Follow the suggestions in the HTML Style Guide
- Validate your code using the HTML Validator (linked in module 0)

Submitting Your Work

Save all files in your CS Unix **public_html/private/10259/a2** directory, which will correspond to the https://csunix.mohawkcollege.ca/~sa#########/private/10259/a2 directory.

Copy all HTML, CSS, and original JavaScript files (not the Bootstrap files) and add “.txt” to the end of the file name of each file. Submit only these files in Canvas – no zips, no images, etc.

Evaluation

See the rubric posted on the LMS for details of the evaluation.

Please direct all questions to your professor, or to the discussion group on the LMS.