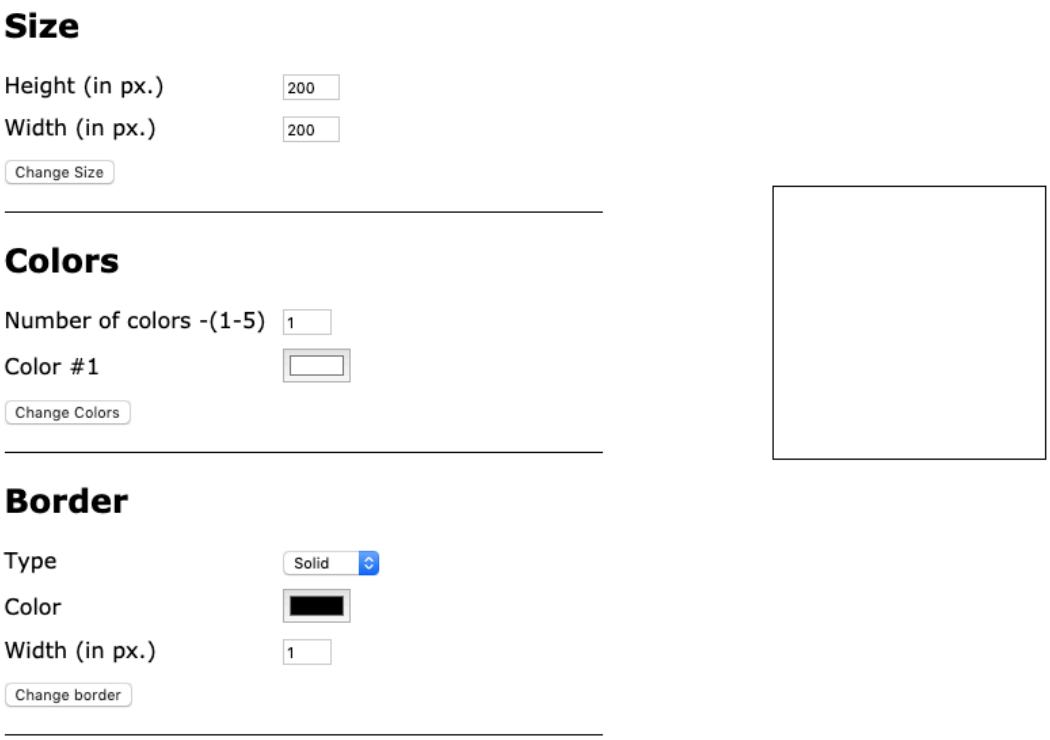
**Assignment 3 – DOM Modification using JavaScript**
**CGS 3066 - Spring 2019**

Deadline: Monday March 11, 2019, 11:59 PM.

**Objective:** Familiarize with DOM Modification using JavaScript

In this assignment, you will complete a dynamic JavaScript enabled styling tool which allows the user to change some CSS properties of a given HTML element, interactively.

**Description:**
Please download and extract the *a3_template.zip* file included with this assignment into your local machine. The compressed file contains an entire website (i.e., HTML, CSS, and some JavaScript code) that needs to be completed. The following is a screenshot of the web site as soon as it is opened in the browser:




The single (and only) web page in the site is divided in two sections: the first one on the left side contains three different menus, and the second one on the right side contains a single div, which acts as the ***output div***. The different menus in the left side modify some of the styling of the *output* div such as its size (width and height), the number of colors displayed inside the *output* div, and the border around it.

Note that as soon as you load the web page, none of the three buttons will work as expected. Your mission then is to implement the necessary functionality **in JavaScript** in order to update the *output* div after the user clicks on any of the three buttons. The expected functionality is described as follows:

**Rubric:** Total: 100 points

1. **Change Size (20 points).** After the user clicks the "Change Size" button, the output div will immediately resize to the dimensions entered in the *Height* and *Width* text boxes.

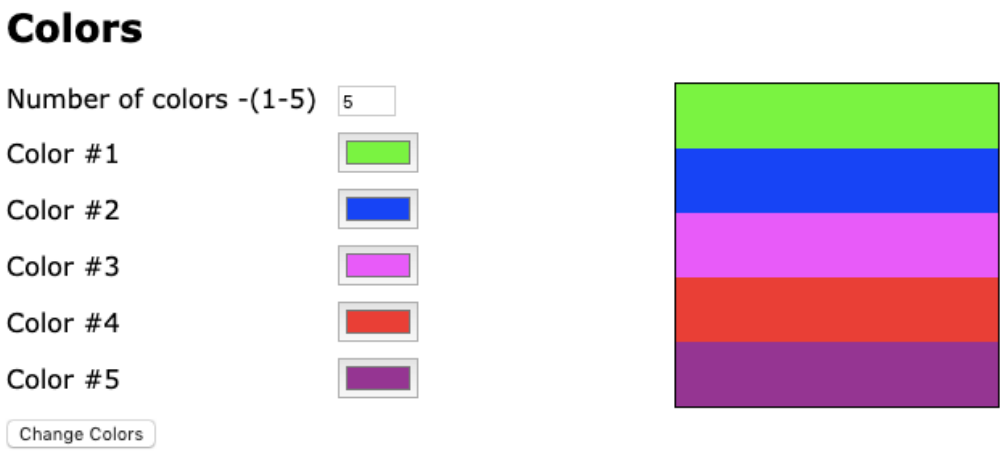
2. **Change Colors (40 points).** The functionality of this menu consists of two steps

   - **Step 1.** The user enters how many different colors need to be displayed in the *output* div. The *Number of colors -(1-5)* box will receive the number of colors.

     Right after the user enters the number of colors, the web page will display as many color pickers as the number that was entered. Each one of the color pickers allows the user to choose a color that needs to be shown in the *output* div.

     **The functionality for Step 1 is already implemented. You do not need to write any additional code to make the colors pickers to show up after the user select the number of colors to be included. The only code you need to write for this functionality the one that implements the next step, Step 2.**

   - **Step 2.** In this step, the user interactively selects as many colors (1-5) as the user wants to display in the *output* div, using the color pickers generated in the previous step. Then, after hitting the *Change colors* button, the web page will display the selected colors inside the *output* div, each one being shown as a horizontal stripe. All the stripes must have the same height (i.e., the *output* div must be equivalently divided to display all the colors). The following screenshot shows an example of an expected output:




   Also implement the **use of negative numbers (-1 to -5) to show vertical stripes instead of horizontal stripes**. That is to say, if the user enters a number between 2 and 5, the *output* div must show horizontal stripes, but if the user enters a number between -2 and -5, the *output* div must show as many vertical stripes as the absolute value of the entered negative number. The width of all the vertical stripes must be the same. (e.g., 2 means two horizontal stripes, -2 means two vertical stripes).

3. **Change Border (30 points).** After the user clicks the "Change border" button, the border for the *output* div will be immediately styled using the *Type*, *Color*, and *Width* indicated in the three menu options.

4. **Code Style (10 points)**
   a. Student's name as a comment at the beginning of each file
   b. Correct indentation and meaningful comments
   c. Correct JavaScript code. Use jslint to validate.

**Rules:**

- All the files included with this assignment (i.e., all the file in the *a3_template.zip* compressed file) are **READ ONLY** except for one: the *scripts.js* file, which is provided to you as an empty file. That is to say, you are not allowed to modify any of the other HTML, CSS, and JavaScript files included with this assignment.
- **Implement all your code inside the scripts.js file.** This is the only file that will be considered for grading.
- No external frameworks/libraries are allowed in this assignment.
- No error checking is needed for input values. You can assume that input values are always valid input values.

**Extra Credit (10 pts).**

Add two buttons at the end of the web page (exact location is up to you). When the user clicks the first button, the *output* div will display an image (you can choose any image you like). When the user clicks the second button, the image must disappear. The image must be strictly smaller than the *output* div (i.e., it should not overlap any side of the *output* div), and the border and size of the *output* div should not be altered in any way. **You must implement everything in JavaScript. The rule about not modifying any other file still applies for this extra credit item.**

**Hints:**

- Even though you are not allowed to modify any of the code included with this assignment, read it carefully. It will contain some hints about how to implement the required functionalities.
- If you want to retrieve the value that each one of the <input> or <select> elements capture (i.e., the input entered by the user), first get a reference to the element using a *querySelector*, and then use the attribute/field **value**. Remember that such value is always returned as a string.

**Submission format:**

Submit to Canvas ONLY the **scripts.js** file.

**Late submission policy:**

As described in the syllabus, any late submission will be penalized with 10% off after each 24 hours late. After 5 days late the grade is 0 (zero).