

Exercise 1: Selecting Elements, Getting and Setting Values

Blue Cow by Gelett Burgess (published in The Lark, 1895)

I never saw a **Blue** Cow, I never hope to see one; But I can tell you, anyhow, I'd rather see than be one.

Change the cow's color to:

Clicking **Change!** does not do anything. That is because our JavaScript and jQuery code has not yet been written.

When the button is clicked, we want all occurrences of the word **Blue** to be changed to the text in the input field, e.g., **Purple**.

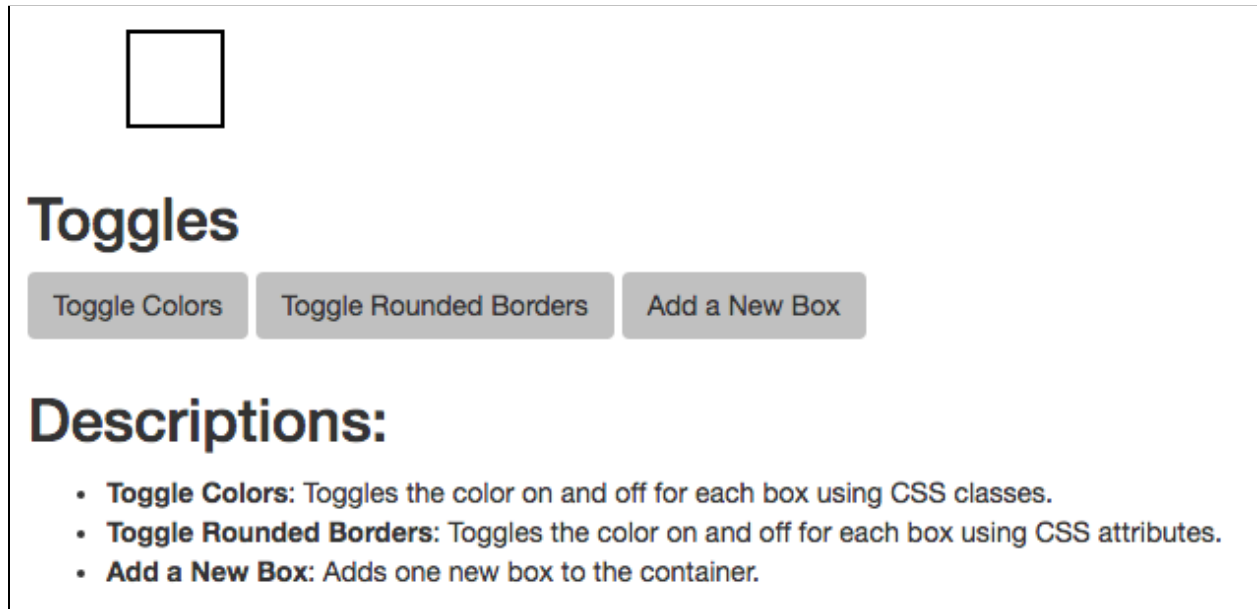
- The **Change!** button has the `id="button_change"`
- The input field has the `id="input_color"`
- All occurrences of the desired text to be replaced are enclosed in `` tags, and have the `class="cow_color"`.

Instructions

You will need to:

1. Select the **Change!** button.
2. Set a click listener on the button to capture click events.
3. Retrieve and store the value currently in the input field.
4. Find and select all occurrences of the relevant text.
5. Replace the text with the stored value.

Exercise 2: Attributes, Classes and CSS Manipulation



The desired behavior for each of the buttons are:

- **Toggle Colors:** Toggles the color between white and cardinal.
- **Toggle Rounded Borders:** Toggles the appearance of a border on the boxes.
- **Add a New Box:** Adds one new box to the container.

Now, open `lab2_ex2.html` in your text editor. Take a moment to understand HTML/CSS code that's present. Note that

- The `<div>` with `id="boxes"` contains three boxes.
- Each of the boxes are of `class="box"`.
- The `<div>` with `id="toggles"` contains three buttons.
 - Button `id="button_toggle_colors"` should toggles the color of the boxes
 - Button `id="button_toggle_roundedges"` should toggle the rounded borders of the boxes
 - Button `id="button_add_box"` should add an additional box

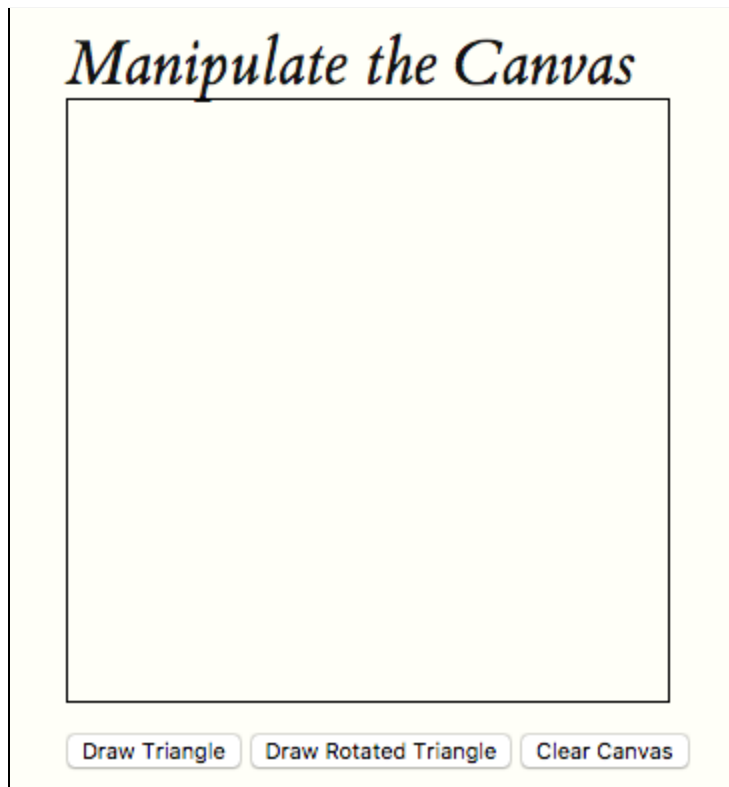
We have already added in the `.click()` listeners for each of the buttons for you. Your job is to fill in the code within those listeners.

Instructions

You will need to:

1. Add the `outlined` CSS class to each of the 3 boxes.
2. Modify the `background-color` CSS attribute on the boxes whenever the **Toggle Colors** button is clicked.
3. Add/Remove the `round-edge` class on the boxes whenever the **Toggle Rounded Borders** button is clicked.
4. Add a new box `<div>` whenever the **Add new box** button is clicked.

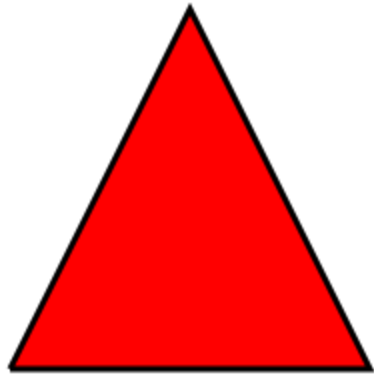
Exercise 3: Canvas Manipulation



Try clicking on any of the three buttons on the page. You will notice that none of them do anything currently. This is because our JavaScript and jQuery code has not yet been put in place. For this exercise, we will be using the HTML5 Canvas. If you are interested in further Canvas documentation, take a look at [HTML5 Canvas Tutorials](#).

The desired behavior for each of the buttons are:

- **Draw Triangle:** Draw a red triangle with black outline onto the canvas.



- **Draw Rotated Triangle:** Draws the same triangle as **Draw Triangle**, but rotated by 45 degrees.



- **Clear Canvas:** Clears the canvas.

Now, open `lab2_ex4.html` in your text editor. Take a moment to understand HTML/CSS code that's present. Note that

- The `<canvas>` with `id="myCanvas"` is 300px by 300px.
 - Button `id="button_draw_triangle"` should draw the triangle with the dimensions specified above.
 - Button `id="button_draw_rotated_triangle"` should draw the triangle with the dimensions specified above, but rotated by 45 degrees
 - Button `id="button_clear_canvas"` should clear the canvas

Instructions

You will need to:

1. Get the context of the `<canvas>` for drawing.
2. Draw a red triangle with black outline when the **Draw Triangle** button is clicked.
3. Draw a rotated red triangle with black outline when the **Draw Rotated Triangle** button is clicked.
4. Clear the `<canvas>` when the **Clear Canvas** button is clicked.