Objectives

- Load an image and draw it on a canvas.
- Access and modify pixel data of the canvas context using the ImageData object.
- Draw a modified ImageData object on a canvas.
- Create and set up a new Web Worker object.
- Post an ImageData object to a Web Worker.
- Receive an ImageData object from a Web Worker.

Description

In this lab you will make use of several of the techniques covered in lecture involving both Web Workers and Image pixel data manipulation. To complete this lab you will need to write at least one HTML file and one JavaScript file. Your HTML file should have at least one <button> element and one <canvas> element.

As soon as your HTML file opens, an embedded script should load an Image file and draw it on the canvas. Images are loaded asynchronously so you should do your drawing as part of the "load" event handler function of the Image. Don't forget to size the canvas to match the dimensions of the loaded image. Also, when the HTML file opens, create one Worker object that preloads your JavaScript file. Refer to examples from the lecture notes.

When the button on your page is clicked, get an ImageData object using a context obtained from your canvas element. Use the `getImageData(…)` method of the context to get its ImageData object. Post a message to your Worker with the ImageData object as message data.

When a new message is received by the Worker, extract the ImageData object from the message event and save a reference to it in a local variable. Loop over all pixel data in the ImageData's Uint8ClampedArray (the ImageData object's `.data` property) and convert all pixels to grayscale. Because the human eye is not equally sensitive to red, green and blue, use the following formula to compute a grayscale value. Reassign all three color channels in the Uint8ClampedArray to this derived grayscale value.

```
grayscale = 0.299*red + 0.587*green + 0.114*blue
```

Return the modified ImageData object by posting it back to the main script. Refer to the examples from lecture on how to modify pixel data in the ImageData object.

In your main script make sure to set up a "message" event handler for your Worker. When a response is received from the Worker, get the modified ImageData object from the event object and draw it on the canvas using the context object's `putImageData(…)` method.

Requirements

- Create at least one HTML and one JavaScript file.
- Please at least one <button> and one <canvas> element on your HTML page.
- When your HTML file opens, immediately load and draw an image on the canvas.
- Also immediately create a new Worker object that preloads your JavaScript file.
- Handle your button's "click" event so that when it is clicked an ImageData object is created from the canvas and passed as message data to your Worker.
- Handle the "message" event inside your Worker script.
- Remove the ImageData object from the event argument passed to the "message" event handler.
- Loop over pixel data in the ImageData's embedded Uint8ClampedArray and convert all to grayscale using the provided formula.
- Post the modified ImageData object back to the main script.
- Handle the Worker "message" event in your main script by extracting the modified ImageData object from the received event
- Draw the modified ImageData object on the canvas element.
- The main script in your HTML file must be enclosed in an IIFE.
- You MUST enter header comments in your JavaScript code including (1) your name, (3) description and or purpose of the assignment.
- You MUST comment your code, explaining what you did in each section.
- Submit JavaScript and HTML files on Canvas under the appropriate assignment.