

Event-Driven Programming

Green Screen Web Page

- Upload and Display an Image

7 min
- Try It! Upload and Display an Image

30 min
- Convert Image to Grayscale

8 min
- Try It! Convert an Image to Grayscale

1h
- Moving to CodePen

9 min
- Try It! Green Screen Online

1h 30m
- Quiz:**
Interactive Web Pages

7 questions
- End of Module Survey

10 min

A PDF copy of this lesson's Try It! exercises can be found in the **Resources** tab.

Develop grayscale algorithm and code at DukeLearnToProgram

1. Make sure you have used the grayscale algorithm to work several examples by hand.
2. Write out the steps of your solution by hand (e.g., loading the image, iterating over each pixel, etc.).
3. Implement the algorithm in the DukeLearnToProgram (DLTP) environment (<http://www.dukelearntoprogram.com/course1/example/index.php>, also linked in the **Resources** tab). Make sure you have a working copy that converts an image to grayscale. It is a good idea to practice any new implementation of a problem in this environment because it has features to help you debug.

Need help? Review the **Convert Image to Grayscale** video and study the algorithm we walk through.

Move code to your web page

1. Fork your previous Pen to start with a web page that allows a user to upload an image, which it will then display.
2. In the HTML panel, add a button that calls the function `makeGray()` when clicked.
3. Add the function *`makeGray()`* to the JS panel. Now you will take the code you wrote in the DLTP environment and adapt it to the web page. Because you are containing the grayscale process in the function **`makeGray()`**, there are some changes to make.

- The grayscale image should be a global variable to allow the **`upload()`** and **`makeGray()`** functions to both access it.
- The function **`upload()`** should set the global image variable to the file input.
- The function **`makeGray()`** should draw the image to the canvas instead of printing it.
- Remember that we are able to use the SimpleImage and SimplePixel libraries because we are importing these libraries in the HTML panel using **`script`** tags.

Need help? Review the code we walk through in the **Convert Image to Grayscale** video; and seek help in the forums!

Refactor your code

Let's restructure and improve on our web page's code so that we have two canvases: one to show the original image and another to show the grayscale version of the image.

1. Add another canvas to the right of the existing one.
2. In the JS panel, declare two global variables, one for the original image, and one for the grayscale image.
3. Modify the function *`upload()`*. When we upload our image file, we want to use it to create two new SimpleImages, one that gets drawn to our canvas on the left and another that we will eventually make gray scale. So:
 - Have **`upload()`** assign a SimpleImage of the uploaded image file to both of the global variables.
 - Also have **`upload()`** draw the original image global variable's Simple Image to the left hand side canvas.
4. Modify the function *`makeGray()`* so that it modifies the grayscale image global variable to grayscale and then displays the modified image in the right hand side canvas.

Need help? If you get stuck, don't give up! You can do this! Review the past couple videos and seek help in the forums.

Mark as completed