

makeBW



This Processing example program, **makeBW**, is designed to convert a color image, like the one on the left above, to a black-and-white image, like the one on the right above. The program is listed below but you should open it in Processing and look at it there.

```
// This program is designed for images that are 900 pixels wide and 600 pixels high.
// For different sizes change lines 4 and 5

int width = 900;
int height = 600;
PImage photo;                                // PImage is an object that contains the information for an image.

void settings()                               // This routine runs once, before the setup routine.
{
  size(width, height);
}

void setup()                                  // The setup routine is run once when the program (sketch )starts.
{
  photo = loadImage("sample.jpg");           // This loads the file "sample.jpg" as an image.
  image(photo, 0, 0);                         // This displays the iamge in the graphics display area.
  photo.loadPixels();                          // Get array of pixel information
}

void draw()                                   // The draw routine is run once for each frame of an animation.
{                                              // This one does nothing but the routine should not be omitted
}

void mouseClicked()                           // The mouseClicked routine is run each time the user clicks the mouse
{                                              // This routine is only needed to respond to mouse clicks.
  color pixel;                                // Used for pixel information.
  float red, green, blue, gray;               // Used for the RGB components of the pixel color
  for(int i = 0; i < width * height; i = i + 1) // Do the following code for each pixel.
```

```

{
    red = red(photo.pixels[i]);      // Get the RGB color values
    green = green(photo.pixels[i]);
    blue = blue(photo.pixels[i]);
    gray = red/3 + green/3 + blue/3; // Use the average for the gray value
    photo.pixels[i] = color(gray, gray, gray); // Change the pixel to gray
}
photo.updatePixels();               // Update the image information with new pixel values
image(photo, 0, 0);                 // Show the black-and-white image
save("sampleBW.jpg");               // Save the black-and-white image as "sampleBW.jpg"
}

```

Run the program. You should see the color image on the first page. Then click anyplace on the image. You should see the black-and-white image on the first page. A copy of the black-and-white image has also been stored in the same folder with the name `sampleBW.jpg`.

All the new ideas can be found in the routine `mouseClicked` that makes the black-and-white image when the user clicks the mouse. The heart of this work is inside the `for` loop:

```

for(int i = 0; i < width * height; i = i + 1) // Do the following code for each pixel.
{
    red = red(photo.pixels[i]);      // Get the RGB color values
    green = green(photo.pixels[i]);
    blue = blue(photo.pixels[i]);
    gray = red/3 + green/3 + blue/3; // Use the average for the gray value
    photo.pixels[i] = color(gray, gray, gray); // Change the pixel to gray
}

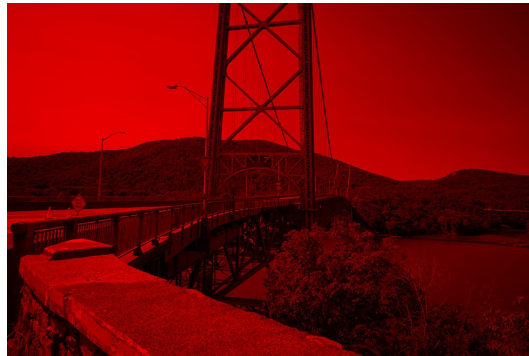
```

The key line computes a gray scale intensity by averaging the red, green and blue intensities. Then the red, green, and blue components of the color for the pixel are all set to the same gray scale intensity value.

The next three lines after this loop update the pixels for the photo image, put the updated photo image in the display area and then save the image in the display area as a jpg file.

Programming Exercises

1. Make a copy of this program in a new folder and put the sample jpg file in the same folder. Rename the folder and program `makeRed`. Then modify `makeRed` to produce an image with just the red component of the color in each pixel. See the image on the next page.



2. Create an image with just the green component.
3. Create an image with just the blue component.
4. Black-and-white photographers often use a collection of filters to create special effects. You can do the same thing digitally. For example, yellow filters remove the blue component from a photograph. This has the effect of darkening the sky and making clouds stand out. Create a black-and-white image by digitally using a yellow filter. The three images below show the effect of a yellow filter. The original color image is on the left, the unfiltered black-and-white image is in the center and the yellow-filtered black-and-white is on the right. (The original jpg is `filter.jpg`). Its width is 600 pixels and its height is 900 pixels.



- 5 Try a red filter.