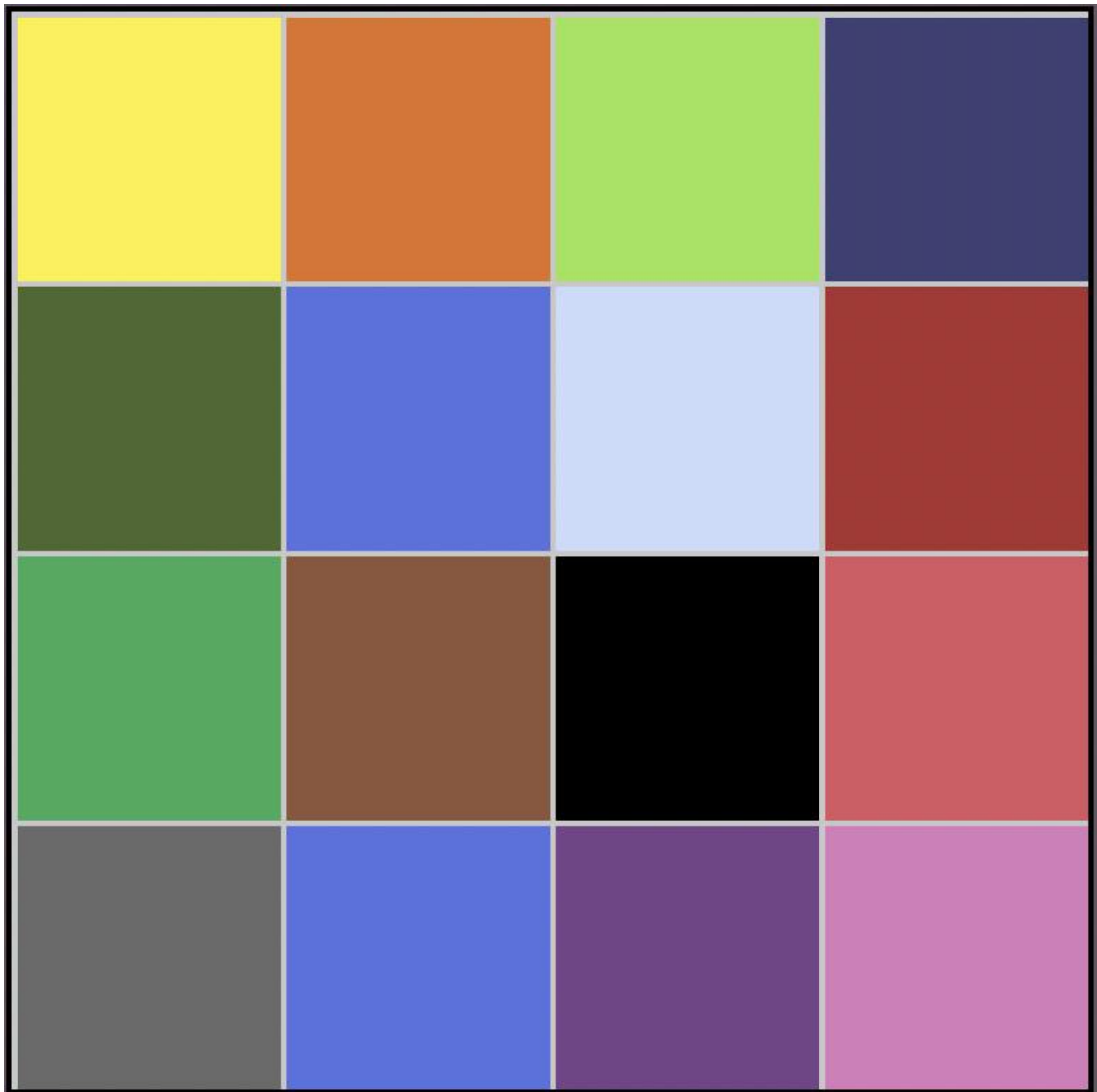


How Does `media.getPixels()` Work?

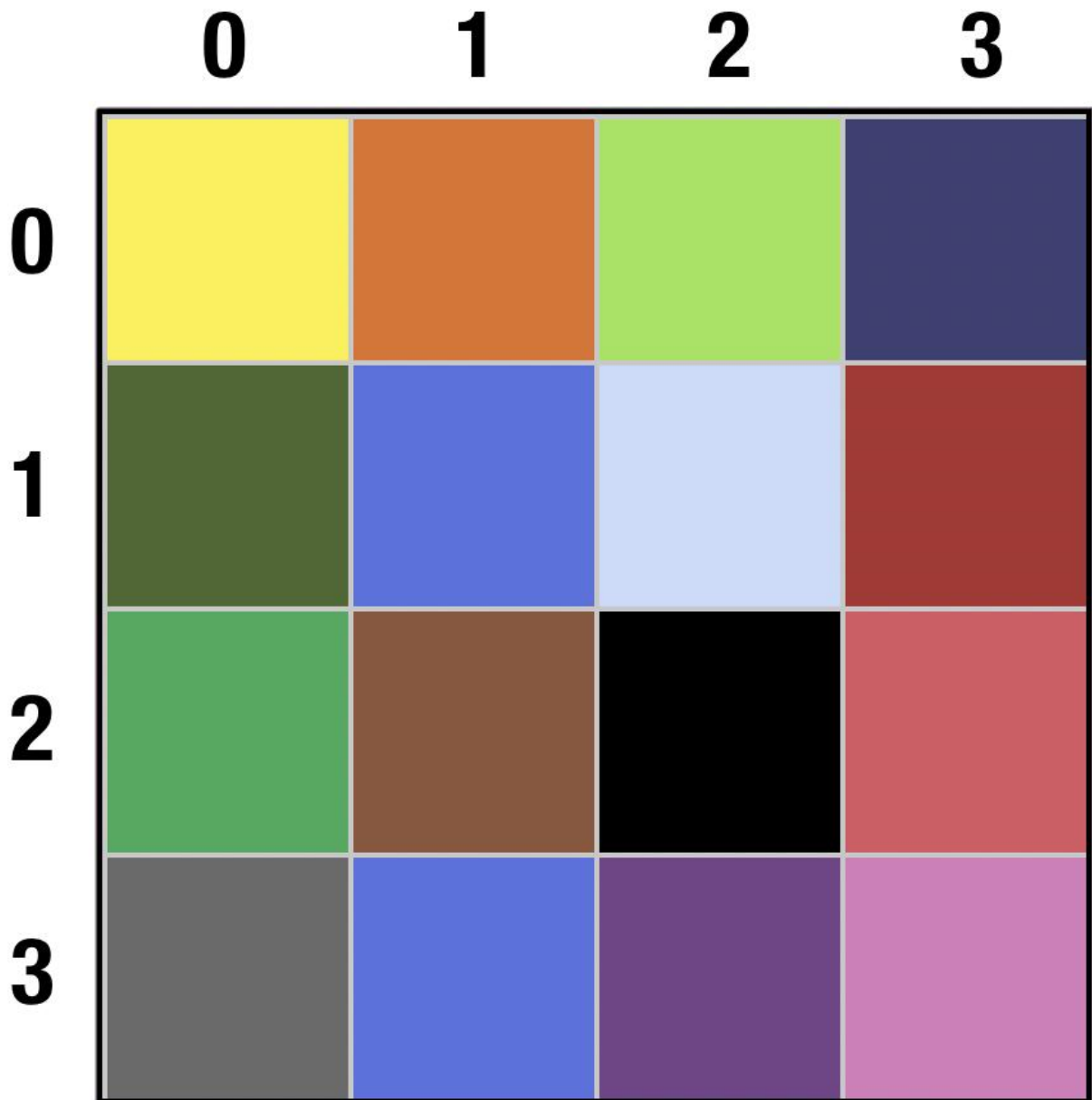
How we digitally store pictures

We know that images are stored as a two dimensional matrix of pixels.

Take the image below:



This is a 4 x 4 image and each pixel can be accessed with a column (x) and row (y) index value.

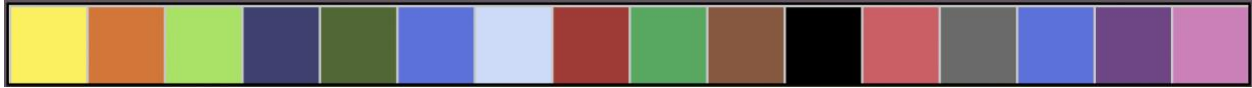


Examples:

- The yellow pixel is in the top-left position of the image at the x and y index positions (0, 0).
- The pink pixel is in the bottom-right position of the image at the x and y index positions (3, 3).
- The brown pixel is located in the image at the x and y index positions (1, 2).

What does `media.getPixels` do?

When we use `media.getPixels` and provide an image as an argument, this function returns a list of pixels where the original image has been flattened into a linear representation. Let's use our 4 x 4 color grid from above.



Notice how all the pixels are still present, but they now appear in a linear ordering matching the two dimensional matrix version of the image in left to right and top to bottom order.

When we want to reference pixels in this orientation, we no longer require a y index position and can access the pixels in the same way we have done previously with characters in strings and lists of string (when we used `split`).



Examples:

- The yellow pixel is at the beginning of the list of pixels at the 0 index.
- The pink pixel is at the end of the list of pixels at the 15 index.
- The brown pixel is located in the list of pixels at the 9 index.

Keep these examples in mind when considering the examples presented in the last few sections of chapter 4 in the textbook.