CSE 120, Section 12
## Important Dates

<table>
<thead>
<tr>
<th>Date and Time</th>
<th>Type</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thursday, 2/13</td>
<td>Checkoff</td>
<td>Color Filters</td>
</tr>
<tr>
<td>Friday, 2/14</td>
<td>Checkoff</td>
<td>Arrays &amp; Elli</td>
</tr>
<tr>
<td>Friday, 2/14 lecture</td>
<td>Quiz</td>
<td>Quiz 3</td>
</tr>
<tr>
<td>Monday 2/17</td>
<td>Vacation :)</td>
<td>No class!</td>
</tr>
</tbody>
</table>
1:1 meetings - sign up on Piazza!
Arrays - like a list of variables
While loops let us update them
Variable scope
Color Filters
File

Make a new folder

Put the image in that folder

Save your processing code for this assignment in the **same** folder

Write down the width and height of photo - pixels
Processing file

Need global variable of type PImage (this goes before setup):

   PImage pic;

color c;

Inside void setup(){} set the size so that it is equal to the width and height of your photo

After that, still inside setup:

   pic = loadImage("greatPic.jpg"); //pic being the variable name
Void draw()
{
  image(pic, 0, 0); // pic being the variable name from the last slide
  loadPixels(); // makes you an array called pixels[]
  c = mouseX + mouseY*width;
  println(int(red(c)) + " " + int(green(c)) + " " + int(blue(c)));
Linearizing a Picture
Linearizing a Picture cont.

Now that you’ve loaded an image into Processing
Linearizing a Picture cont.

The grid represents the pixels in the image, so each grid square represents one pixel.
Linearizing a Picture cont.

- **loadPixels()**
  - Loads a snapshot of what is currently on your canvas into an array `pixels[]`

- **pixels[]**
  - An array that stores all the values of the pixels on the canvas
  - Stores values of the `color` datatype
Linearizing a Picture cont.

pixels | color (255,0,255) | 15
---|---|---
0 | | 0

The diagram shows a grid of pixels with varying colors, indicating different linearization processes.
Linearizing a Picture cont.

What does `pixels[6]` return?

What does `pixels[15]` return?

What does `pixels[16]` return?
Color Filters

keyPressed() function - if statements to check if ‘r’ is pressed

Then refill pixels with `pixels[i] = color(red(pixels[i]), 0, 0);` with a loop

draw() { updatePixels();}
Restore

keyPressed() - for not ‘r’

  Redraw picture  - image()

  - loadPixels()
More colors

Repeat for red with

- ‘g’ green
- ‘b’ blue
- ‘c’ green + blue -> cyan
- ‘y’ red + green -> yellow
- ‘m’ red + blue -> magenta
  - REMINDER - red =
    - keyPressed() function - if statements to check if ‘r’ is pressed
    - Then refill pixels with pixels[i] = color(red(pixels[i]), 0, 0); with a loop
Checkoff
Worksheet! (?)
Friday office hours?
Worktime

- Color Filters
- Arrays & Elli
- Quiz studying