Quick Review Summary

Reference Summary List

Brent M. Dingle, Ph.D. 2015
Game Design and Development Program
Mathematics, Statistics and Computer Science
University of Wisconsin - Stout

Lecture Objectives

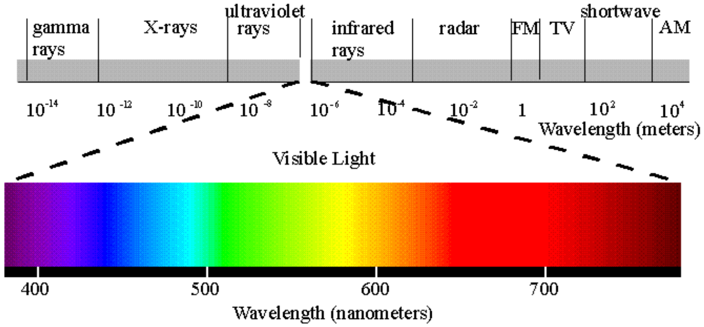
- Quick review of topics covered and direction headed

List Summary

- Previously
  - What a Digital Image is
    - Acquisition
    - Human Perception
    - Representation
  - Color Spaces
  - HTML5 and JavaScript Code Examples
    - Pixel manipulation
    - Image Loading
    - Filtering

What is a Digital Image?

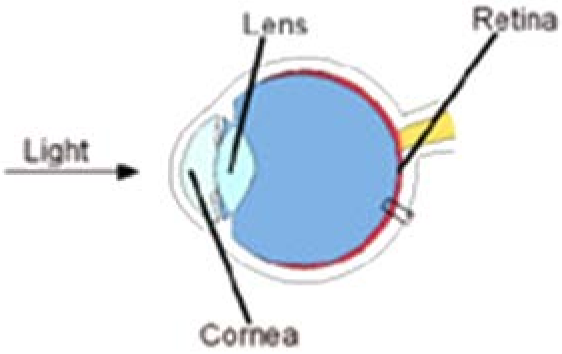
- **Acquisition** Sources
  - Electromagnetic (EM) spectrums
  - Acoustic Imaging
    - EX: Ultrasounds, Seismic Imaging…
  - Electron Microscopy

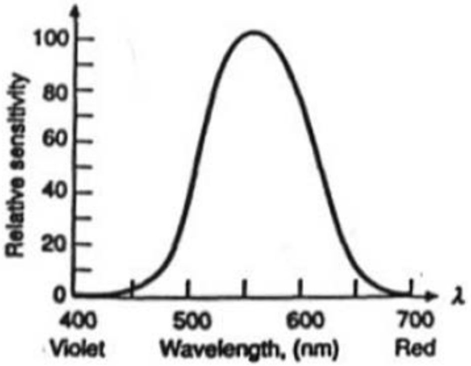


- **Generation**
  - Synthetic Images
    - Computer Generated
      - Graphics, Games, Digital Art…

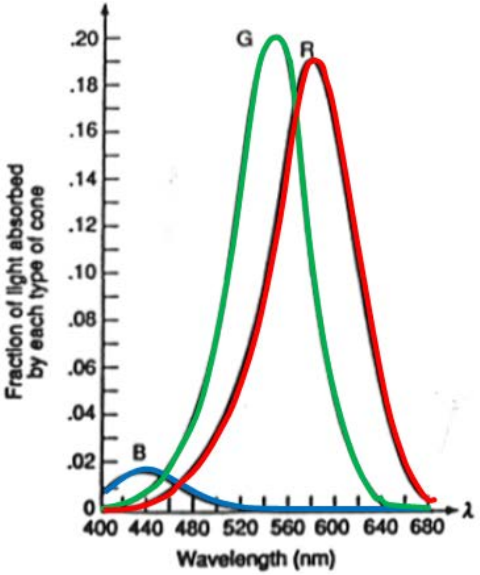
What is a Digital Image?

- **Human Perception**
  - Human Eye is limited





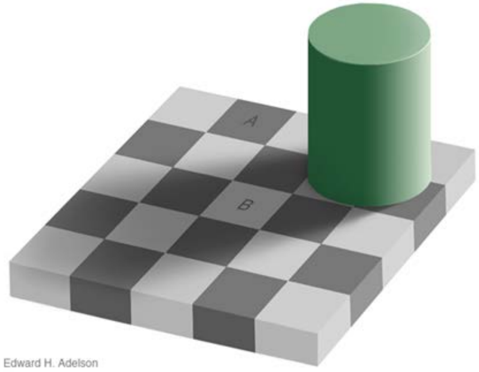
Rods: 75-150 million, all over the retina surface and sensitive to low levels of illumination

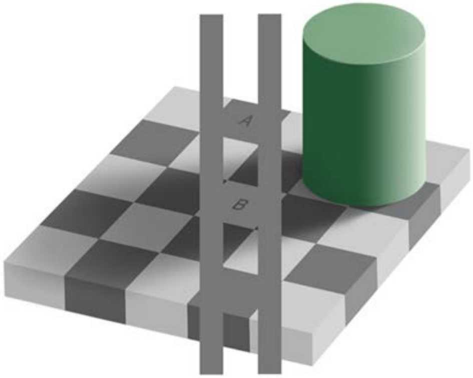


Cones: 6-7 million in each eye, central part of retina (fovea) and highly sensitive to color

What is a Digital Image?

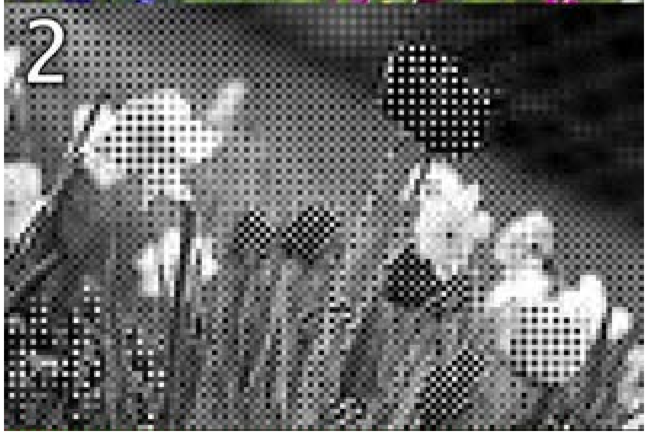
- **Human Perception**
  - Human Brain interprets and adjusts

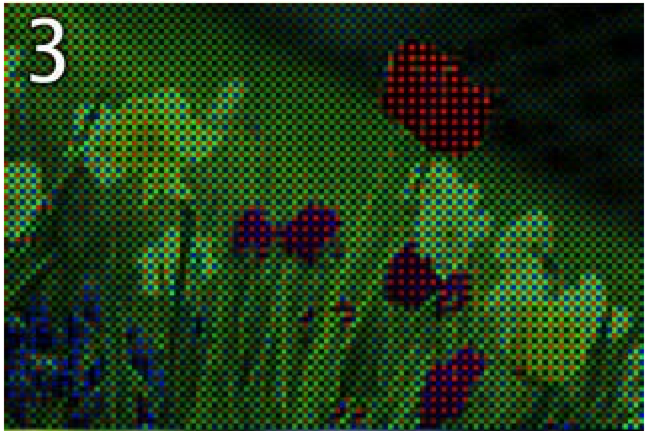




What is a Digital Image?

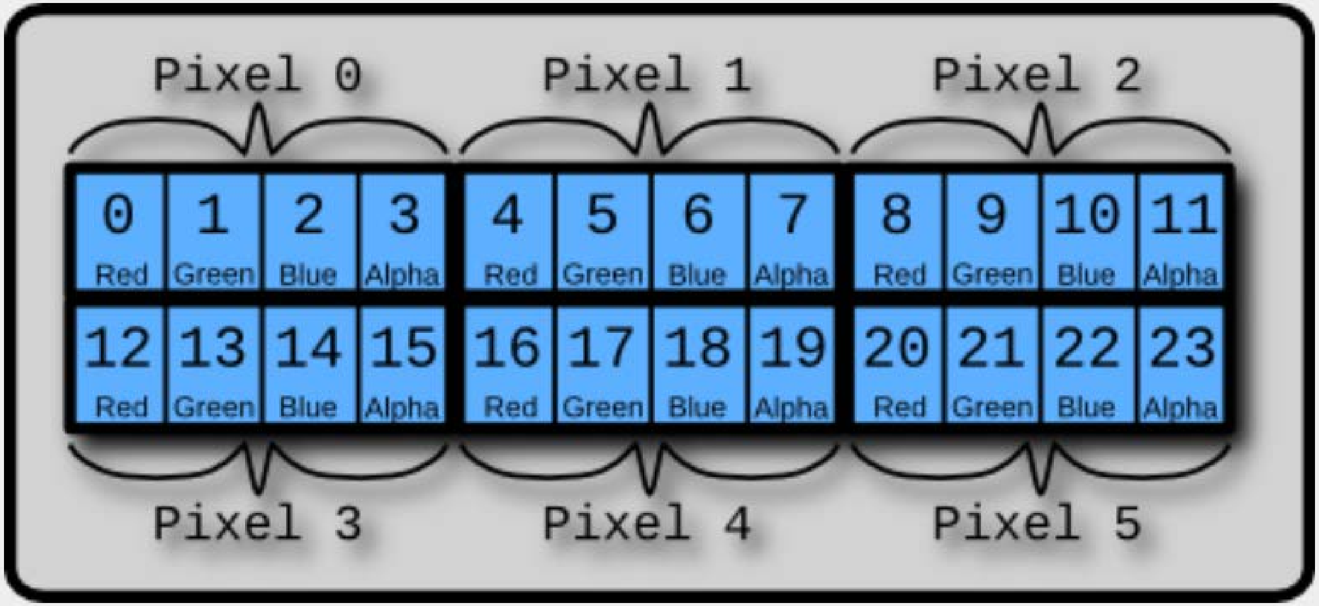
- Representation
  - – Is designed for humans
    - Red, Green, Blue→ **hardware based acquisition**





What is a Digital Image?

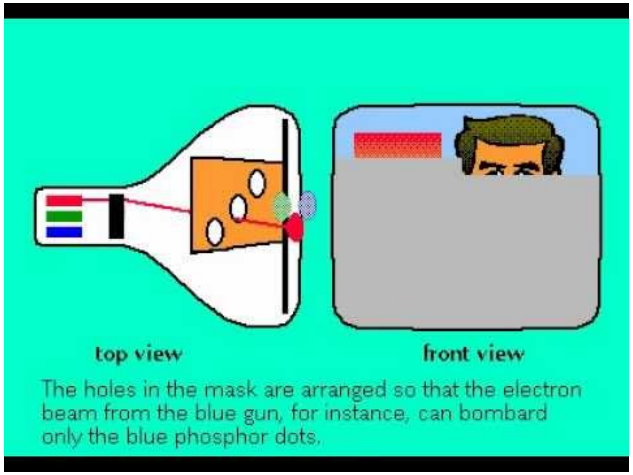
- Representation
  - – Is designed for humans
    - Red, Green, Blue→ **storage**



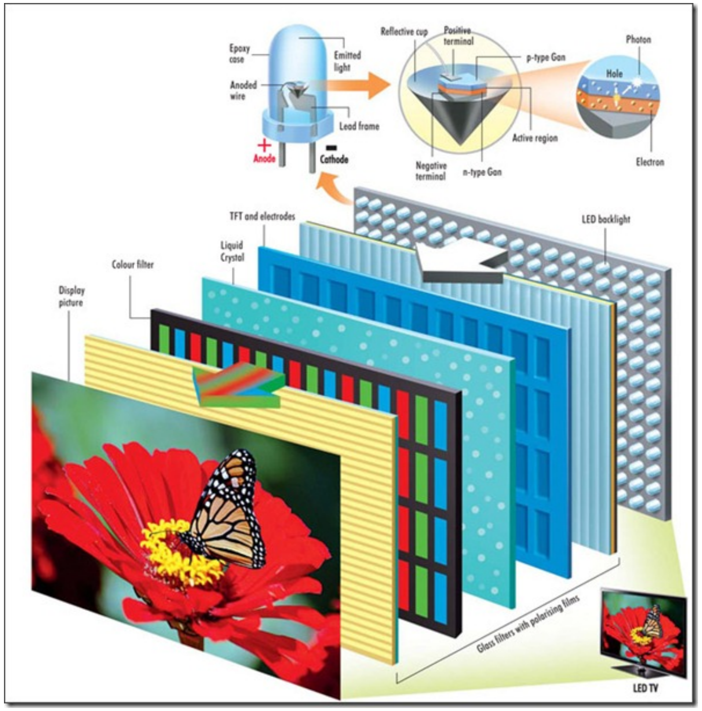
Pixel layout in the pixel array for a 3-by-2 image of 6 pixels. Each pixel takes 4 elements in the array for red, green, blue, and alpha, for a total of 24 array elements, 0-23.

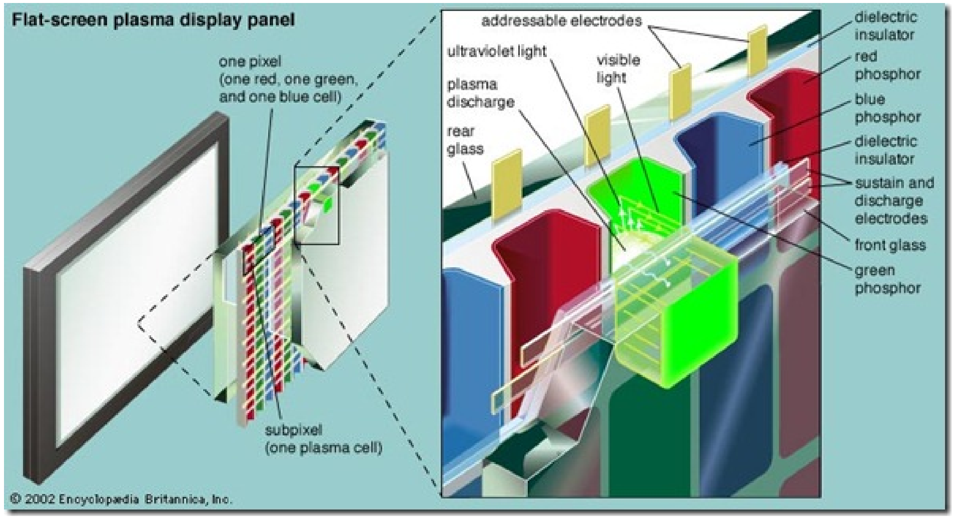
What is a Digital Image?

- Representation
  - – Is designed for humans
    - Red, Green, Blue→ **display**



The holes in the mask are arranged so that the electron beam from the blue gun, for instance, can bombard only the blue phosphor dots.





© 2002 Encyclopædia Britannica, Inc.

http://www.guidingtech.com/26940/led-lcd-plasma-difference/

List Summary

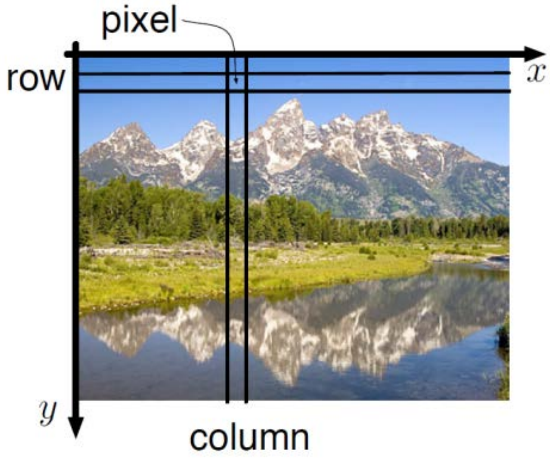
- Previously
  - What a Digital Image is
    - Acquisition
    - Human Perception
    - Representation
  - Color Spaces
  - HTML5 and JavaScript Code Examples
    - Pixel manipulation
    - Image Loading
    - Filtering

Color Spaces

- Our images are designed for
  the human eye
  and
  the human brain
  - How does that work?

Color Spaces

- We have seen
- Representation is Red, Green, Blue
- Digital Storage is 2D Array of Pixels
  - Each Pixel with R, G, B values



- So…

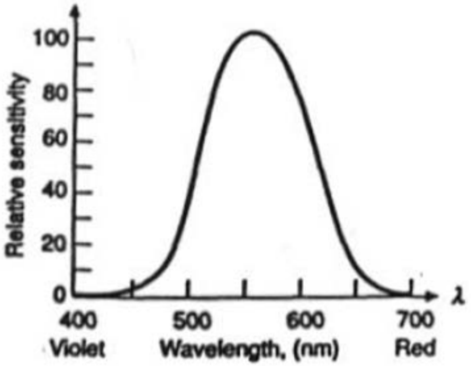
Color Spaces

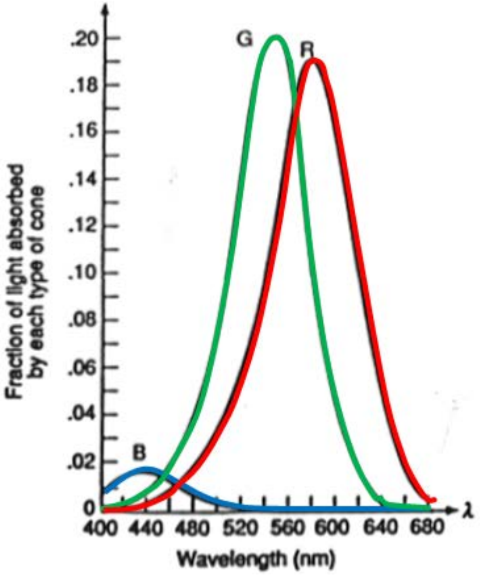
- What's the theory?
  - How is the representation defined?

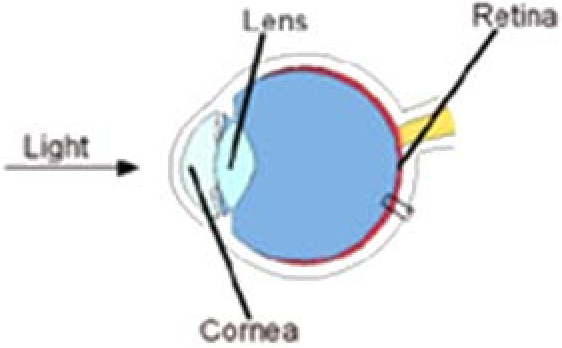
- **Tri-Stimulus Theory of Color**

Color Spaces: Human Eye

- **Study the Human Eye**



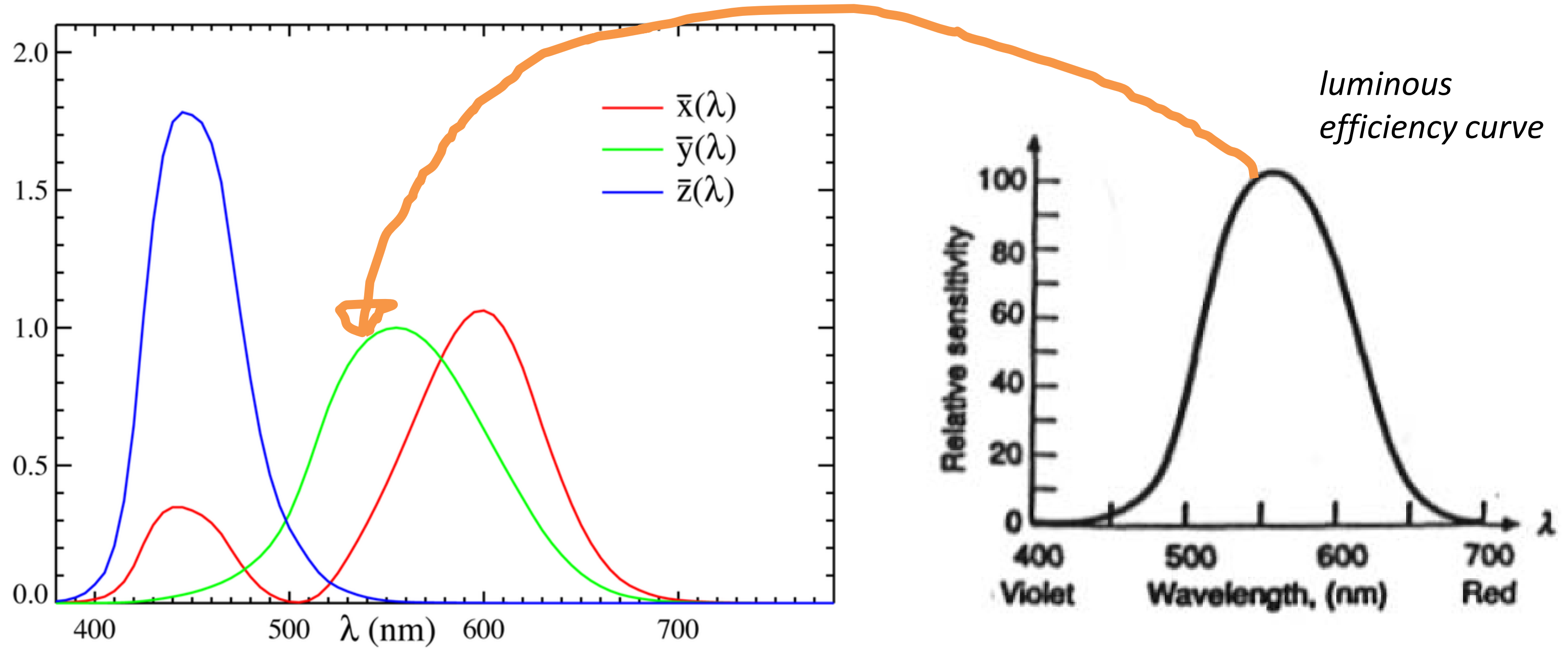




Rods: 75-150 million, all over the retina surface and sensitive to low levels of illumination

Cones: 6-7 million in each eye, central part of retina (fovea) and highly sensitive to color

CIE Color Matching Functions



luminous efficiency curve

- **DESIGN a model**
  - By design the y curve is just the luminous efficiency curve of the human eye

Tri-Stimulus Theory of Color

- **Make the model work**
  - *Easy to use with hardware and humans*
  - *Easy to transition to equivalent models*
    - *Experiment results extend across models*
    - *So digital image (color) representation, use, perception becomes measurable, comparable, consistent*

    - Additive Color Systems
      - RGB
      - HSV
      - CIE xyY

    - Subtractive Color System
      - CMY
        - » CMYK

List Summary

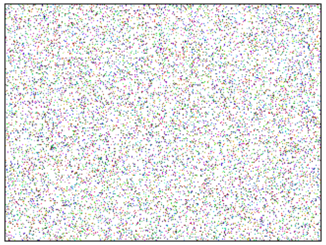
- Previously
  - What a Digital Image is
    - Acquisition
    - Human Perception
    - Representation
  - Color Spaces
  - HTML5 and JavaScript Code Examples
    - Pixel manipulation
    - Image Loading
    - Filtering

Math and Programming

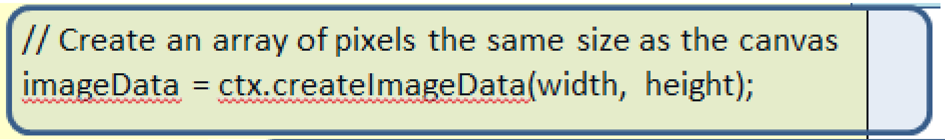
- Established is a model that is capable of acquiring, storing, and displaying images
  - It is mathematical in nature and based on experimental results of human perception

- This provides a “world” to play in
  - Mathematical theory and algorithmic exploration is fun
  - Programming computers based on/using such theory and experimentation is needed to “see” results

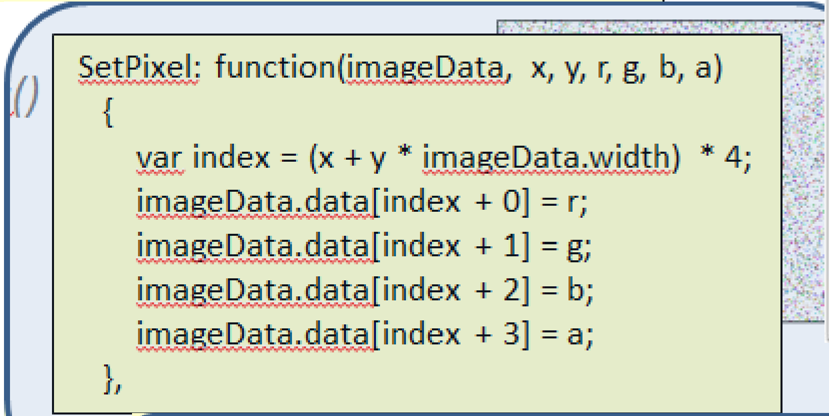
Coding Examples



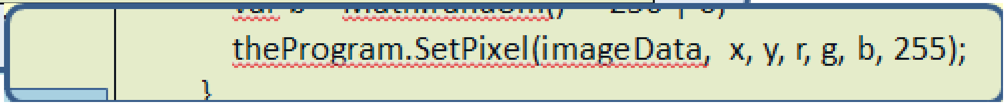
- HTML5 and Javascript support digital image manipulation
  - Pixel based access to images through arrays



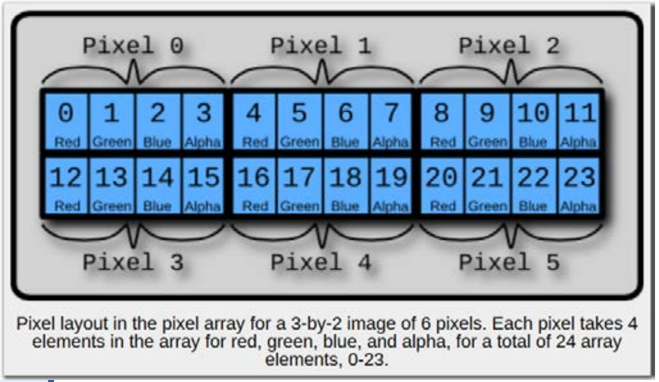
```
// Create an array of pixels the same size as the canvas
imageData = ctx.createImageData(width, height);
```



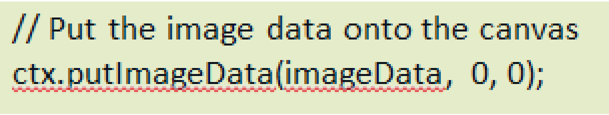
```
SetPixel: function(imageData, x, y, r, g, b, a)
  {
     var index = (x + y * imageData.width) * 4;
     imageData.data[index + 0] = r;
     imageData.data[index + 1] = g;
     imageData.data[index + 2] = b;
     imageData.data[index + 3] = a;
  },
```



```
        theProgram.SetPixel(imageData, x, y, r, g, b, 255);
     }
```



Pixel layout in the pixel array for a 3-by-2 image of 6 pixels. Each pixel takes 4 elements in the array for red, green, blue, and alpha, for a total of 24 array elements, 0-23.



```
// Put the image data onto the canvas
ctx.putImageData(imageData, 0, 0);
```

Coding Examples

- Loading an Image
  - Operations exist to decompress and load images via javascript through the browsers

```
// Setup the listeners
      this.dropArea = document.getElementById('theDropArea');
      this.dropArea.addEventListener('dragover', this.onDragOver, false);
      this.dropArea.addEventListener('drop', this.onDropFileSelect, false);
```

```
onDropFileSelect: function (evt)
   {
      // get array of filenames that were dragged
      var files = evt.dataTransfer.files;

      // If the "first" file is not an image, do nothing
      var curFile = files[0];
      // Only process image file
      if ( curFile.type.match('image.*') )
      {
         var img = new Image;
```

```
   img.onload = function()
   {
      var canvas = document.getElementById(theProgram.SOURCE_IMG_CANVAS_ID);
      var ctx = canvas.getContext('2d');
      canvas.style.display = "block";
      canvas.width = img.width;
      canvas.height = img.height;
      canvas.style.width = canvas.width + "px" ;
      canvas.style.height = canvas.height + "px";
// Can draw the image on the canvas
      ctx.drawImage(img, 0, 0);
// And can store the image data into a data structure
      theProgram.srcData  = ctx.getImageData(0, 0, img.width, img.height);
   }
   img.src = URL.createObjectURL(curFile);
```

Coding Examples

- Having stored the image in a data structure we can write code to apply mathematical operations to the image
  - e.g. convolution filtering

$$I\_new(x,y) = \sum_{j=-1}^{1}\sum_{i=-1}^{1} \alpha_{ij} I\_old(x-i, y-j)$$

```
// -----------------------------------------------------------------------
   applyConvIdentity: function()
   {
      // below should do 'nothing' → applies a filter but changes nothing
      var destData = theFilter.convolute(theProgram.srcData,
                                   [ 0,  0,  0,
                                     0,  1,  0,
                                     0,  0,  0   ]);
      theProgram.displayOutput(destData, theProgram.srcData);
   },
```

k(j, i) indexes into the convolution filter

i.e. k(j, i) is [[0,0,0],[0,1,0],[0,0,0]]

f(x, y) index into the image

i.e. f(x,y) is referring to theProgram.srcData

theFilter.convolute(f[][], k[][])

```
for y = 0 to imageHeight
    for x = 0 to imageWidth
        sum = 0
        for i = -1 to 1
            for j = -1 to 1
                sum = sum + k(j+1, i+1) * f(x – j, y – i)
            end for j
        end for i
        g(x, y) = sum
    end for x
end for y
```

List Summary

- Previously
  - What a Digital Image is
    - Acquisition
    - Human Perception
    - Representation
  - Color Spaces
  - HTML5 and JavaScript Code Examples
    - Pixel manipulation
    - Image Loading
    - Filtering

- Near Future
  - Image Manipulation
    - Filtering
    - Enhancement
    - Convolutions

Questions?

- Beyond D2L
  - Examples and information can be found online at:
    - *http://docdingle.com/teaching/cs.html*
    - *Continue to more stuff as needed*

Extra Reference Stuff Follows

Credits

- Much of the content derived/based on slides for use with the book:
  - *Digital Image Processing,* Gonzalez and Woods

- Some layout and presentation style derived/based on presentations by
  - Donald House, Texas A&M University, 1999
  - Bernd Girod, Stanford University, 2007
  - Shreekanth Mandayam, Rowan University, 2009
  - Igor Aizenberg, TAMUT, 2013
  - Xin Li, WVU, 2014
  - George Wolberg, City College of New York, 2015
  - Yao Wang and Zhu Liu, NYU-Poly, 2015
  - Sinisa Todorovic, Oregon State, 2015