

Today's "Menu"

(week 10)

- Announcements
- Mostly Images ...
- Questions?

31-Mar-10

COMP 480 - Winter 2010

1

Announcements

31-Mar-10

COMP 480 - Winter 2010

2

Images



31-Mar-10

COMP 480 - Fall 2009

3

Pixels

- **Pixel** – picture element
- Pixel contains **8-bit RGB** values
- **8 bits** => $2^8 = 256$ patterns, range 0 – 255
- **255** – max color value, 0 – zero color value ...
- **RGB** tuples (**rvalue**, **gvalue**, **bvalue**)

(255, 0, 0) - red	(0, 0, 127) - ?
(0, 0, 0) - ?	(0, 255, 255) - ?
(255, 255, 255) - ?	(127, 127, 127) - ?

31-Mar-10

COMP 480 - Fall 2009

4

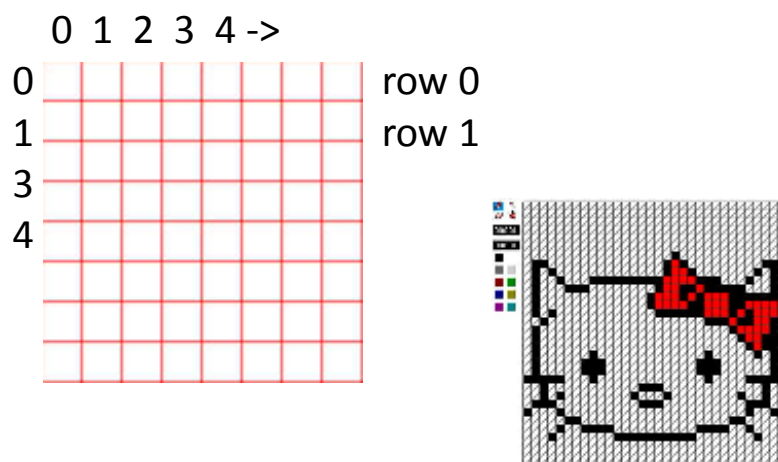
Images

- Image made up of pixels (e.g., 580x240)
- Common image formats: .jpg, .gif, .bmp, .png
- Image is a 'grid' or '2D array'. Each pixel has an 'address' in row/column values

31-Mar-10

COMP 480 - Fall 2009

5



31-Mar-10

COMP 480 - Fall 2009

6

PIL

```
import Image

im = Image.open('pic.jpg')
print 'image size: %d x %d' %(im.size[0], im.size[1])
print 'mode: %s    format: %s' %(im.mode, im.format)
im.show()
```

- More info - google “python pil api”
- **API** - **A**pplication **P**rogramming **I**nterface

31-Mar-10

COMP 480 - Fall 2009

7

PIL

```
import Image

im = Image.open('pic.jpg')
print('image size: %d cols x %d rows' %
      (im.size[0], im.size[1]))
print 'mode: %s    format: %s' %(im.mode, im.format)
im.show()

pix_ar = im.load()      # load image into 2D array
red_pixel = 255, 0, 0    # new red RGB pixel
cols = im.size[0]
rows = im.size[1]
```

- More info - google “python pil api”
- **API** - **A**pplication **P**rogramming **I**nterface

31-Mar-10

COMP 480 - Fall 2009

8

Things to try ...

- Exercises/Questions
 - How to draw a vertical or horizontal line?
 - Diagonal?
 - Reverse Diagonal?
 - Surround image with border?
 - Save your image?
 - 'pepper' the image 'randomly'
- More info, google 'python pil api'

31-Mar-10

COMP 480 - Fall 2009

9

```
import Image

im=Image.open('pic.jpg')

pix_ar = im.load()      # load image into 2D array
red_pixel = 255, 0, 0    # a red RGB pixel
rows = im.size[1]
cols = im.size[0]

# draws a 'diagonal' line
for row in range(rows): # our nested loop
    for col in range(cols):
        if row == col:
            pix_ar[col, row] = red_pixel

im.show()
```

31-Mar-10

COMP 480 - Fall 2009

10

Summary/Recap

- Questions?