**matplotlib pyplot**

We import matplotlib.pyplot as plt when we want to look a the pixels in a file.

We read in a file doing this:

img1 = plt.imread(someFileName)

Now someFileName is a pixelated version of your picture. We can manipulate each pixel in it by walking through the rows and columns of the picture.

some of the commands we used with numpy:

To set up a screen with images on it

fig,ax = plt.subplots(rows of images,columns of images)

So, for example if you want 6 pictures on the screen 3 across the top and 3 across the bottom:

fig,ax = plt.subplots(3,2)

Each pixel has an amount of red ([row][column][0]), an amount of green ([row][column][1]), and an amount of blue ((row][column][2]).

Example: if we want to take all of the **blue** out of a picture and replace it with black

for r in range(picSize):

for c in range(picSize[0]):

img[r][c][2] = 0;

Other commands we can use with matplotlib:

axis('on') imshow(img)

axis('off') set\_title(String)

set\_xlim(xmin,xmax) set\_ylim(ymin,ymax)

set\_xlabel(String) set\_y\_label(String)

**PIL**

We import PIL if we want to manipulate an entire picture like resize it. We read PIL files in like this:

img = PIL.Image.open(filename)

Then we can do things like resize, etc. Also, **PIL.ImageDraw**

open() paste() Draw()

new() transform() ellipse()

crop() save() line()

convert() rotate() polygon()   
resize() text()

If you are using the same image and want to go back and forth to do things to pixels and whole image:

Example:

pyplotImage = plt.imread(filename)

pilImage = PIL.Image.open(fileName)

Go from the plt image 🡪 pil image: img2 = PIL.Image.fromarray(pyplotImage)

Go from the pil image 🡪 plt image: img3 = np.array(PIL.Image.open(filename)