Example: simple image editing

images are represented as a rectangular matrix of pixels. Inspect properties of an image on your computer and you'll see the dimension expressed as, e.g. 600 x 400 pixels. This means 400 rows of 600 pixels. 782 pixels wide

tall



Dimensions: 782×1124 Color space: RGB Color profile: Color LCD Alpha channel: Yes Last opened: Today 10:58 AM 1124 pixels UNWORLDLY LI ORMAN DUSH'S NEW YOM

- A common way to represent color is using combination of three components: red, green, blue. Called RGB color.
 - Use a tuple, with first component expressing red, second expressing green, third blue.
 - Some systems use 0.0->1.0 for range. "Standard" or common is integers 0->255. You'll hear things like "24 bit color" or "8 bits per channel"
 - In 0->255 system:
 - (0,0,0) is black
 - (0,0,255) is bright blue
 - (255,255,255) is white
 - (255,255, 0) is combination of "full" red and "full" green, which is ?? (I always forget ...)
 - Easy to find interactive color charts/tools on the web ...
 - Note that there are other color representations/systems RGB isn't the only one. CMYK, HSV, ...

- Not important for this example but something to think about...
- How much storage does it take to represent an image?
- For instance given what I said on previous page, how much memory (bytes/megabytes) does it take to store a 1024 x 1024 RGB image?
- Now look at the properties of some photographs stored on your computer – compare pixel dimensions and disk storage space used. Does the numbers make sense to you? Can you explain the relationship?
- Similarly, how much space might a video say a 2-hour movie – require? (HD video is 1920 x 1080) When you download a movie, how big is file? Do the numbers make sense?

- To write a simple image editing program in Python, you only need to understand
 - basics of rgb color
 - basics of image coordinates
 - how to retrieve color values from image getpixel
 - how to set color values in image setpixel

Set pixel at pos=(column,row) on image, with color=(r,g,b). def setpixel(image, pos, color):

```
r,g,b = color
```

column,row = pos

image.put("#%02x%02x%02x" % (r,g,b), (column, row))

```
column, row = pos
stringvalue = image.get(column,row)
colortuple = tuple(map(int, stringvalue.split()))
return(colortuple)
```

- look at simpleImageManip.py
 - Easy to make grayscale, negative, flipped, redder images
- for fun, simpleSteganography.py

easy to hide secret messages in images