Section 4

May 13, 2021

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Today’s Agenda

- Social Time (5 minutes)
- Recap/New (10 minutes)
- Section Filter problem (15 minutes)
- Trim Crop problem (15 minutes)
- Questions/comments/discussion (5 min)

Where We Are At

- Images lectures began last Friday
- Text processing began yesterday
- Assignment 3 will be released Monday
- Lessons 1-10 have been published

Social Time

- How was your week?
- Triumphs? Challenges?
- Any aha moments?
- Any new ideas for how to use Python?

Python Standard and Third-Party Libraries

- In addition to the official docs...
- Automate the Boring Stuff with Python (free online)
- Python Crash Course (includes cheat sheets)

Automate the Boring Stuff with Python

- In addition to Python fundamentals...
- File manipulation
- Web scraping
- Working with Excel, Google Sheets, PDFs, Word Docs, CSV, and JSON data
- Task scheduling
- Sending emails and text messages
- Image manipulation
- Controlling keyboard/mouse through GUI manipulation

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Python Crash Course

- In addition to Python fundamentals...
- Game project
- Data Science project
- Web Development project
- Plus cheat sheets

Recap: Image Manipulation

- Images (pixels, axis, RGB)
- SimpleImage library
- Iteration: for-each loop
- Iteration: for-each loop versus nested for loop
- getting/setting pixels
- Handy tips/commands

Recap: Pixels and Axis

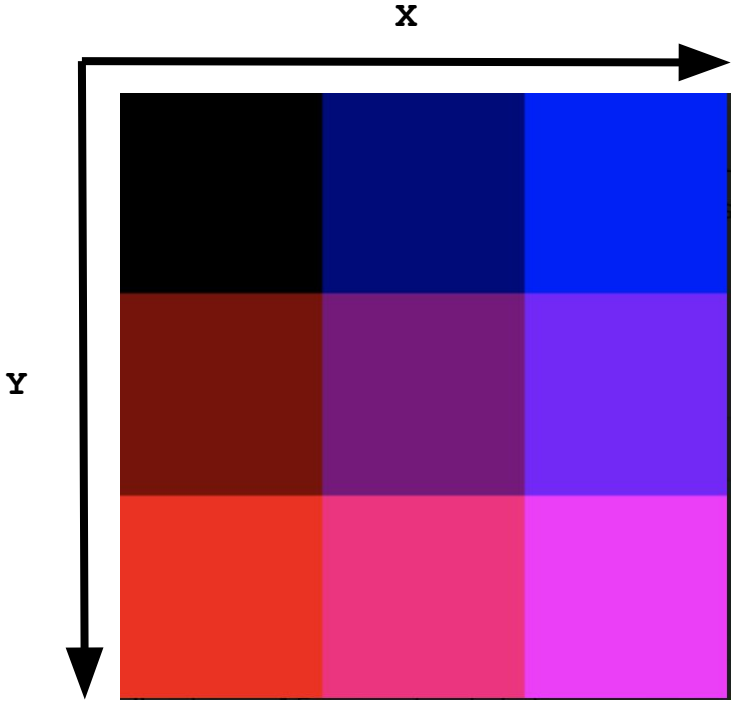
An image is a grid of square pixels.

We can draw an x and y axis along our image.

An image has a height and a width:

```
image.height
image.width
```

Adapted from a walkthrough by Akshay Jaggi!



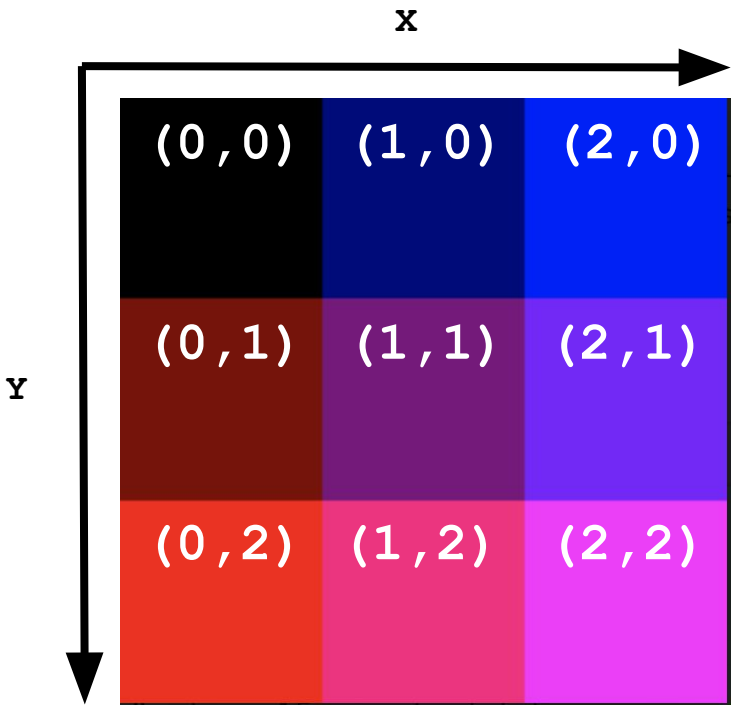
Katherine Michel, Section Leader, Stanford Code in Place, 2021

Recap: x and y Coordinates

This means that every pixel has x and y coordinates:

pixel.x
pixel.y

y increases going down
x increases going right

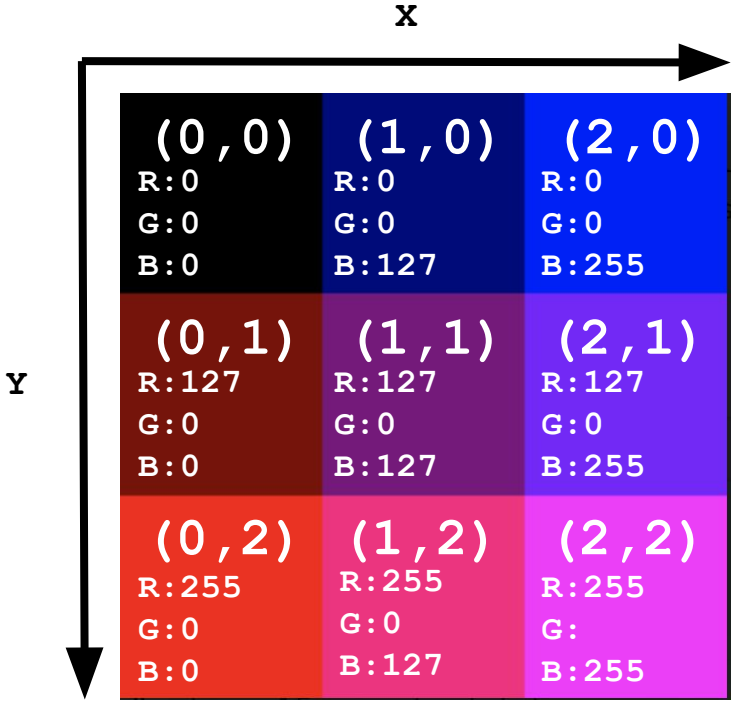


Recap: RGB

Along with a location, every pixel has 3 RGB values which represent the brightness of that color:

pixel.red
pixel.green
pixel.blue

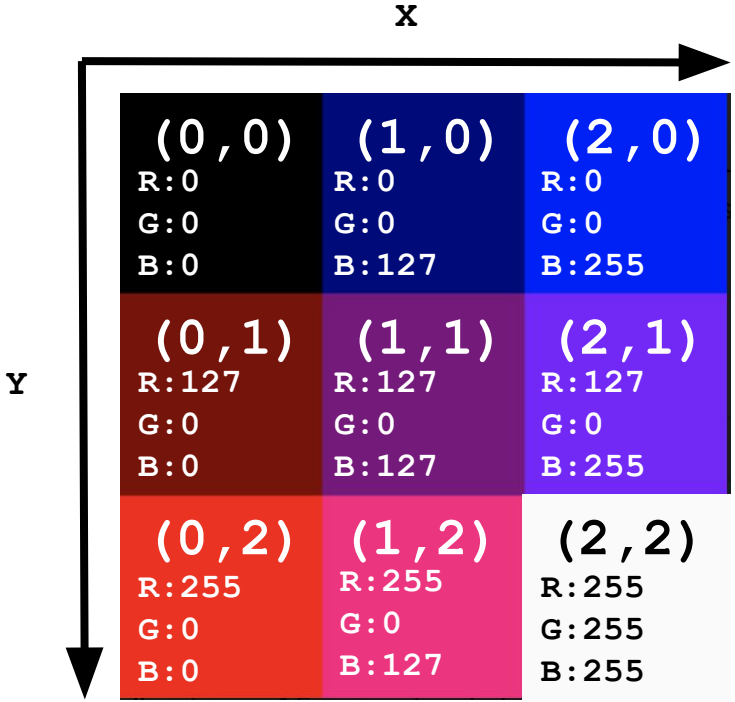
The values of each color are represented as integers between 0 and 255



Katherine Michel, Section Leader, Stanford Code in Place, 2021

Recap: RGB

The effect of adding green to the final square results in white (presence of all color)



Katherine Michel, Section Leader, Stanford Code in Place, 2021

SimpleImage Library: Accessing Data

Dimension: image.height

image.width

Color: pixel.red

pixel.green

pixel.blue

Location: pixel.x

pixel.y

Image and pixel data can be accessed using "dot notation"... this is common Python

Recap: for-each Loop

Common Python syntax

```
for item in items:
    # do something
```

Used to iterate through items in an iterable/sequence (list, tuple, string)

SimpleImage for-each loop

```
from simpleimage import SimpleImage

image = SimpleImage("image.jpg")
for pixel in image:
    # iterate through each pixel
```

Recap: for-each Loop Versus Nested for loop

for-each Loop

```
for px in image:
    do_something(px)
```

Use this if you don't need to access the coordinates

Nested for loop

```
for x in range(image.width):
    for y in range(image.height):
        px = image.get_pixel(x, y)
        do_something(px)
```

Use this if you need to access the coordinates

Recap: for-each Loop Versus Nested for loop

for-each Loop

```
for lala in image:
    do_something(lala)
```

Nested for loop

```
for x in range(image.width):
    for y in range(image.height):
        lala = image.get_pixel(x, y)
        do_something(lala)
```

“pixel” variable can actually be anything
Python recognizes that “for” “something” “in”... means iterate
Similar to parameters (Python understands position)

Recap: Getting/Setting Pixels

You can reset the color value:

```
for pixel in image:
    pixel.red = pixel.red // 2
```

Locations cannot be reset. *If you want to move a pixel:* Create a blank image and set pixels in new image.

Access a pixel from coordinates:

```
pixel = image.get_pixel(x, y)
```

You can set the value of a pixel:

```
image.set_pixel(x, y, pixel)
```

Recap: Integer Division

- We are dealing with integers, not floats
- We need to use floor division (//) not (/)
- With floor division, we get an integer

Right

```
px.red = pix.red // 2
```

Wrong

```
px.red = pix.red / 2
```

Recap: Handy Commands

Import module:	from simpleimage import SimpleImage
Read an image **from a file**:	image = SimpleImage(“image.jpg”)
Create a blank image:	image = SimpleImage.blank(width, height)
Resize to target image:	image.make_as_big_as(target_image)
Show image:	image.show()

Norak Filter Problem

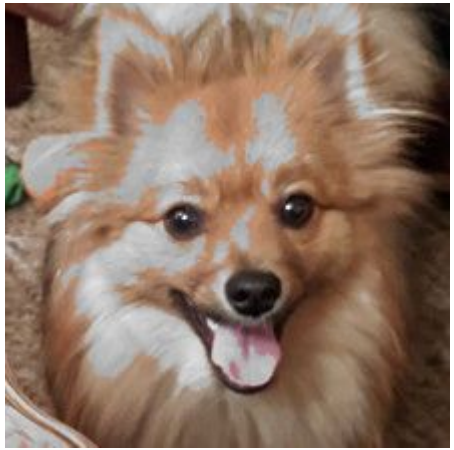
Replace each bright pixel in an image with its grayscale equivalent. A bright pixel in an image is defined as a pixel whose average component is greater than 153 (which is 60% of 255). All non-bright pixels in the image should remain unchanged.

To make a pixel grayscale, set each of its red, green and blue components to be equal to its average component.

Tip! Define a function to get the average component of a pixel (this is a repeated task).

Before

After

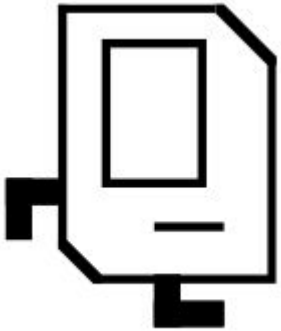


Katherine Michel, Section Leader, Stanford Code in Place, 2021

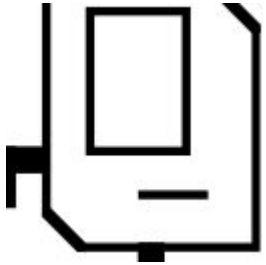
Trim Crop Problem

Write a function trim_crop_image which removes trim_size amount of pixels from each side of the image.

Before: call trim_crop_image on this picture of Karel



After: remove trim_size = 30 pixels from all sides



Both width and height have been reduced by 60 pixels

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Add Border Problem

Add a black border to an image.

Before:

add a border of size
border_size = 10

After

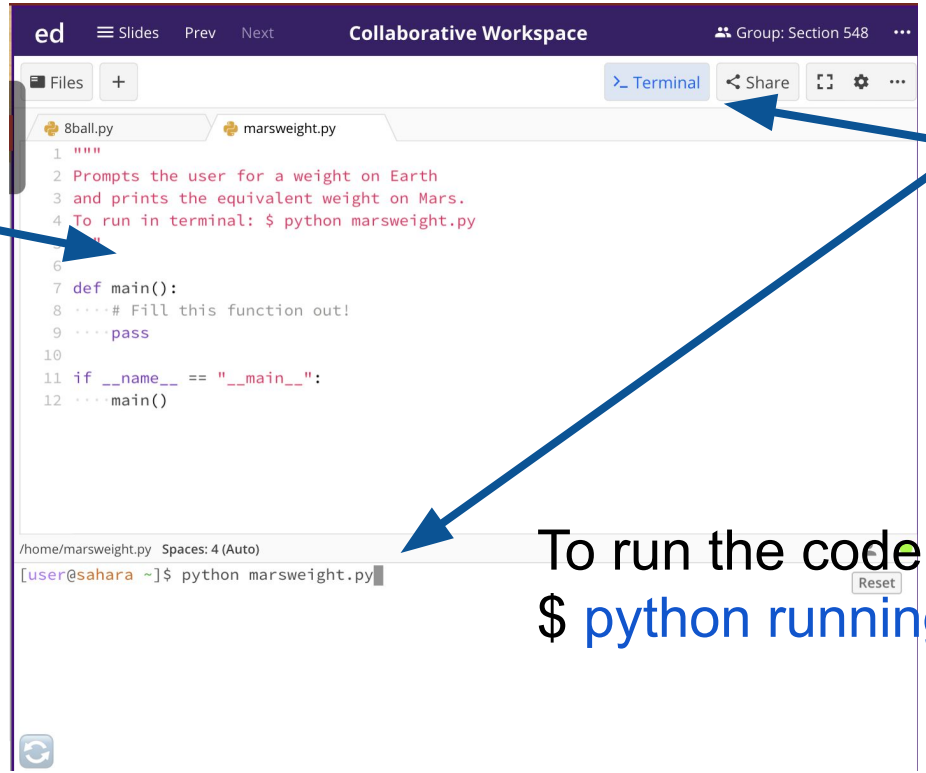
Both width and height have been reduced by 20 pixels

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Closing… Thoughts, Questions, Discussion?

Collaborative Workspace

Everyone can see the same code and edit collaboratively



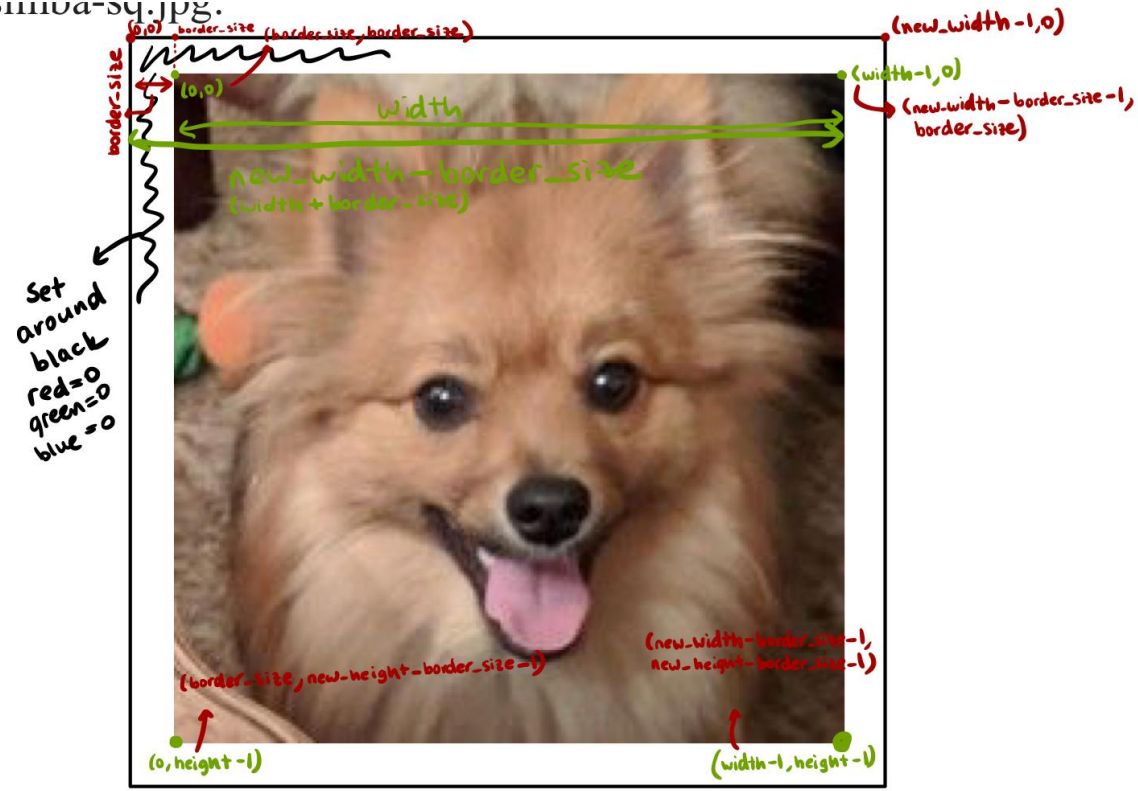
The “Terminal” button can be used to open a terminal…

To run the code type file name
$ python running.py

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Add Border Problem

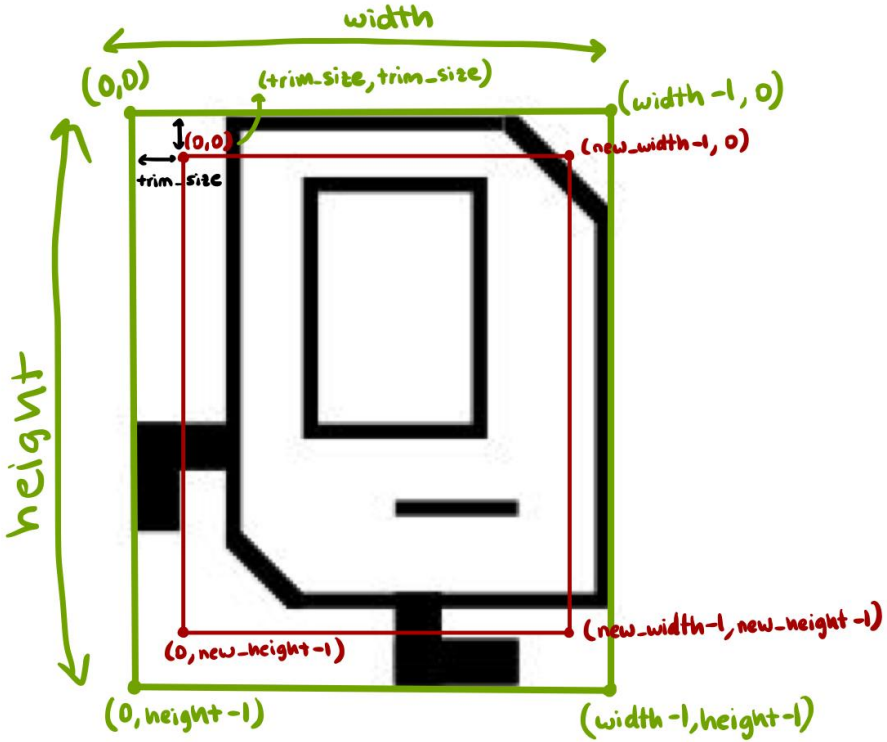
simba-sq.jpg:



Drawing by Aybüke!

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Trim Crop Problem



Drawing by Aybüke!

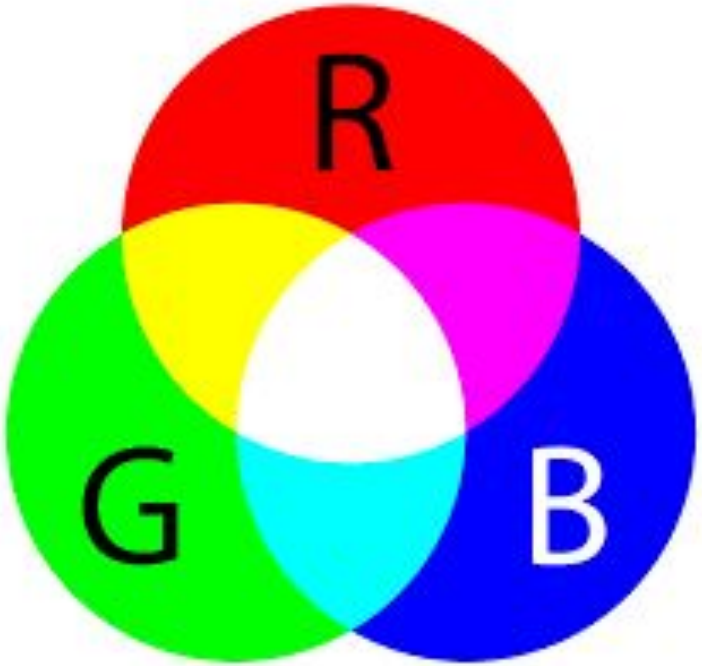
Katherine Michel, Section Leader, Stanford Code in Place, 2021

Recap: RGB

Each pixel color has 3 RGB values which represent the brightness of that color

R = Red
G = Green
B = Blue

The values of each color are represented as integers between 0 and 255



Katherine Michel, Section Leader, Stanford Code in Place, 2021