

# 15-110: Principles of Computing, Spring 2018

## Lab 11 – Thursday, April 12

### Goals

In previous labs you've seen how to draw simple shapes on the canvas and combine them to make interesting graphics like fractals. In this lab you will continue with graphics, exploring how arbitrary images can be encoded as two-dimensional lists of pixels (where each pixel is itself a list) and manipulated using iteration.

When you are done you should be able to do the following:

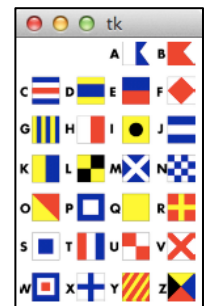
1. Understand how a two-dimensional list of pixels maps to the image on the screen.
2. Create a list representing a simple image.
3. Be fluent with code handling two-dimensional lists.
4. Perform simple geometric transformations on an image representation, like flipping horizontally or vertically.
5. Perform simple color transformations on an image representation.

### Part 1: TA Demonstrations (Displaying images in Python)

Students should create a lab11 directory. Then download the files `image.py` and `nautical.ppm` to this directory and `cd` to this directory. Run `python3` and display the nautical flag alphabet on a Canvas by evaluating the following Python expressions:

```
>>> from image import *
>>> img = read_ppm("nautical.ppm")
>>> c = create_canvas(img)
```

Students should look at the image itself to see that it is a list of lists of lists (rows, columns, pixels). Each pixel is a 3-element list consisting of red, green and blue intensities.



In the subsequent problems, students will write functions to take an image (a list of lists of lists) as its parameter, modify the image and return it as the result. Then the `draw_image` function can be used to draw the updated image on the canvas `c`:

```
>>> img = image_update_function(img)
>>> draw_image(c, img)
```

## Activities

1. Write a function `flip_image(img)` in the file `flip_image.py` that flips image `img` upside-down and returns the resulting image. To flip the image upside-down, you need to flip the rows of the image: the first row swaps with the last row, the second row swaps with the second-to-last row, etc.

Hint: If `L` is a list, `L.reverse()` will reverse the elements of the list. If the elements of the list are lists, the lists are reversed, but the contents of the lists are not reversed. For example:

```
>>> L = [[1,2,3],[4,5,6],[7,8,9]]
>>> L.reverse()
>>> L
[[7, 8, 9], [4, 5, 6], [1, 2, 3]]
```

Note that the `reverse` function modifies the original list. It does not return anything.

Use the `flip_image` function to return the flipped version of the nautical flag image. The `draw_image` function comes from the `image.py` file.

Sample usage:

```
python3 -i flip_image.py
>>> from image import *
>>> img = read_ppm("nautical.ppm")
>>> c = create_canvas(img)
>>> img = flip_image(img)
>>> draw_image(c, img)
```

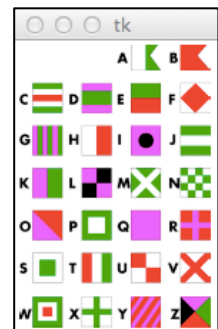


2. Write a function `swap_bg(img)` that swaps the blue and green values of every pixel in the image `img`, and returns the resulting image. Test it on the nautical flag alphabet image. What happens if you call `swap_bg` repeatedly on the same image?

Algorithm: For each row and column in the image, you have a pixel (a list of three integers). Take the last two values of that list and swap them.

Sample usage:

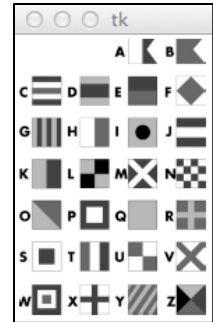
```
python3 -i swap_bg.py
>>> from image import *
>>> img = read_ppm("nautical.ppm")
>>> c = create_canvas(img)
>>> img = swap_bg(img)
>>> draw_image(c, img)
```



3. Write a function `grayscale(img)` in the file `grayscale.py` that converts an image `img` to grayscale by setting each pixel's `r`, `g`, and `b` values to their average value  $(r+g+b) // 3$ , and returns the resulting grayscale image. Test your function on the nautical flag alphabet image. NOTE: Make sure you use integer division.

Sample usage:

```
python3 -i grayscale.py
>>> from image import *
>>> img = read_ppm("nautical.ppm")
>>> c = create_canvas(img)
>>> img = grayscale(img)
>>> draw_image(c, img)
```

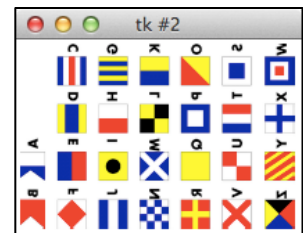


4. Write a function `transpose(img)` in the file `transpose.py` that creates a new image by flipping the given image `img` along the major diagonal, so that the rows become columns and the columns become rows. The function should return the resulting new image.

NOTE: For this problem, you will need to display the resulting image on a new canvas in a new window. Do you see why?

Sample usage:

```
python3 -i transpose.py
>>> from image import *
>>> img = read_ppm("nautical.ppm")
>>> c = create_canvas(img)
>>> img2 = transpose(img)
>>> c2 = create_canvas(img2)
```



## Submission

When you finish the lab, you should be inside the `lab11` folder, which is inside the `private/15110` directory. When you type `ls` and press the Enter key, you should see one or more of the following files: `flip_image.py`, `swap_bg.py`, `grayscale.py`, and `transpose.py` along with `image.py` and the `nautical.ppm` image. Type `cd ..` to move up one folder and press the Enter key. Then, zip your `lab11` folder by typing `zip -r lab11.zip lab11` and you should see a `lab11.zip` file in the current folder if you type `ls`. Please submit the zipped file `lab11.zip` on Autolab under 'lab 11'.