Computer Science 1 — CSCI 1100
Lab 4 — Images and Modules

This lab explores the use of images and modules. We strongly urge you to reference Lecture 7:

`https://www.cs.rpi.edu/~mushtu/CS1100/lecture_notes/lec07_modules_images.html`

You can also find more details on all image functions at:

`http://pillow.readthedocs.org/en/latest/handbook/tutorial.html`

In this lab, you have three checkpoints (the third checkpoint has two options, you can do either of them). Before you start, we recommend that you verify your pillow installation. Open your Spyder IDE and go to your python shell pane. Enter:

```
from PIL import Image
im = Image.new('RGB', (200,200))
im.show()
```

If you get any errors, it is likely that you did not get `pillow` properly installed during Lab 0. Please visit:

`https://www.cs.rpi.edu/~mushtu/CS1100/python_environment.html`,

search for `conda install pillow`, and follow the directions provided.

Checkpoint 1: Two by two wallpaper

To start this part, first download the `lab04files.zip` file from Submitty. The file contains many images. Unzip this folder in your Dropbox folder for Lab4. For simplicity, make sure that your programs are saved in the same directory as these images before you try to execute them. Otherwise, you will get a `File Not Found` error when trying to open the images.

In this first part of the lab, you are going to create a two-by-two wallpaper using any four of the images. We recommend images titled `'ca.jpg','im.jpg', 'hk.jpg','bw.jpg'`, but you can use any image you wish. For this lab, you are going to hard code the names of the images into your program.

Open a new file and save it out as `check1.py`.

Next, create a `512x512` blank document. Then, open each image, resize it to size `256x256` and paste them into this new image to form a 2x2 wallpaper.

All functions to do this are illustrated in the examples in the Lecture 7 notes. And, remember, the paste function modifies an image but does not return anything.

When done, call the `show()` function to check that the wallpaper looks correct. You will see that the images are distorted because the original images are not square and the resize does not preserve the aspect ratio. We will fix this issue in for `Checkpoint 2`.

For your convenience, here are a few image functions that might help you:

```
from PIL import Image    ##must import Image first to be able to use it
im = Image.new(mode, size, color)  # use mode 'RGB', color 'white'
im = im.resize((width,height)) # resize to the given width/height passed as a tuple
im.paste(pasted_image, (x,y)) # (x,y) coordinates of upper left corner as a tuple
```

```
im.save(filename)
im.show()
```

To complete Checkpoint 1, show a TA or a Mentor your code and the output of your program.

Checkpoint 2: Image correction

To start this part, you are going to create a new file called `check2_helper.py` and create a function in it called `make_square`. Your function takes as an argument an image object, crops it to make it square and returns the resulting image. Remember to import Image in your file.

In your function, you will check the size of the image and crop some part of it to make it into a square image. If the image is wider than longer (horizontal), then you must crop it along the x-axis. If the image is longer than wider (vertical), then you must crop it along the y-axis. Always start from the top left corner.

Check the course notes for finding the size of an image and cropping. Remember that the crop function returns a new image.

Now write some simple code to use this function to make one of the images from first checkpoint square and show. We recommend you try this with some of the images titled `inc*` as well to see if it works for horizontal images as well as vertical ones.

Here is how example test code should look:

```
im = Image.open('1.jpg')
imsquare = make_square(im)
imsquare.show()
```

You see that your function does not open an image, but takes an image as input argument, and returns a new image. Once you are convinced that your function works, leave nothing in the file but the function.

Copy your solution from checkpoint 1 to a new file called `check2.py`. Modify your code so that:

- You now import your own module `check2_helper.py`
- You use your own function from this module to make each image in the wallpaper square before resizing them.

To complete Checkpoint 2, show a TA or a Mentor your code and the output of your program. We will check that you are calling your module correctly, and that your program has correct structure.

Checkpoint 3:

Please come to lab for the last checkpoint.