1. What does RGBA stand for?

RGBA stands for red, green, blue, and alpha (transparency) in the color

An RGBA value is a tuple of 4 integers, each ranging from 0 to 255. The four integers correspond to the amount of red, green, blue, and alpha (transparency) in the color.

2. From the Pillow module, how do you get the RGBA value of any images?

import the Image module of PIL into the shell:

>>>from PIL import Image

create an image object and open the image for reading mode:

>>>im = Image.open(‘myfile.png’, ‘ r’)

myfile is the name of the image to be read and give the appropriate file format. that is if its a jpeg image then give it as myfile.jpg

we use a function of Image module called getdata() to extract the pixel values. this scans the image horizontally from left to right starting at the top-left corner. The values got from each pixel is then added into a list. Finally what we get is a list with each pixel value as a set of 4 values(R,G,B.A).

>>>pix\_val = list(im.getdata())

pix\_val is the list that contains all the pixel values which can be printed to see those values But the list got is a list of sets and some times its needed to flatten the list for example if the list is like: [(123,124,145,120), (345,453,234,124),……] and the list that is needed is [123, 124, 145, 120, 345, 453, 234, 124….] then the command to flatten the list is:

>>> pix\_val\_flat = [x for sets in pix\_val for x in sets]

This list comprehension extracts each element of each set in the list pix\_val and all the elements are stored in pix\_val\_flat.

3. What is a box tuple, and how does it work?

The box.tuple submodule provides read-only access for the tuple userdata type. It allows, for a single tuple: selective retrieval of the field contents, retrieval of information about size, iteration over all the fields, and conversion to a Lua table.

Below is a list of all box.tuple functions:

**Name** **Use**

box.tuple.new() Create a tuple

box.tuple.is() Check whether a given object is a tuple

#tuple\_object Count tuple fields

box.tuple.bsize() Get count of bytes in a tuple

tuple\_object[field-number] Get a tuple’s field by specifying a number

tuple\_object[field-name] Get a tuple’s field by specifying a name

tuple\_object[field-path] Get a tuple’s fields or parts by specifying a path

tuple\_object:find(), tuple\_object:findall() Get the number of the first field/all fields matching the search value

tuple\_object:next() Get the next field value from tuple

tuple\_object:pairs(), tuple\_object:ipairs() Prepare for iterating

tuple\_object:totable() Get a tuple’s fields as a table

tuple\_object:tomap() Get a tuple’s fields as a table along with key:value pairs

tuple\_object:transform() Remove (and replace) a tuple’s fields

tuple\_object:unpack() Get a tuple’s fields

tuple\_object:update() Update a tuple

tuple\_object:upsert() Update a tuple ignoring errors

4. Use your image and load in notebook then, How can you find out the width and height of an Image object?

from google.colab import files

uploaded = files.upload()

from PIL import Image

myImg = Image.open('abc.jpg')

w,h = myImg.size

w,h

5. What method would you call to get Image object for a 100×100 image, excluding the lower-left quarter of it?

ImageObject.crop((0, 50, 50, 50))

6. After making changes to an Image object, how could you save it as an image file?

By Calling the imageObj.save('new\_filename.png') method of the Image object.

7. What module contains Pillow’s shape-drawing code?

The ImageDraw module contains code to draw on images

8. Image objects do not have drawing methods. What kind of object does? How do you get this kind of object?

ImageDraw objects have shape-drawing methods such as point(), line(), or rectangle(). They are returned by passing the Image object to the ImageDraw.Draw() function