1. What does RGBA stand for?- Red Green Blue Alpha
2. From the Pillow module, how do you get the RGBA value of any images?

* import the Image module from the Pillow library. from PIL import Image.
* Open any image and get the RAGBAG values. img = Image.open('image.png') rgba = img.convert(“RGBA”) ...
* Change the color. Data will be an Imaging Core object containing thousands of tuples of RGBA values. ...
* Store the changed image.

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

The box.tuple submodule provides read-only access for the tuple userdata type.

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.

1. Use your image and load in notebook then, How can you find out the width and height of an Image object?
2. # import required module
3. from PIL import Image
5. # get image
6. filepath = "geeksforgeeks.png"
7. img = Image.open(filepath)
9. # get width and height
10. width = img.width
11. height = img.height
13. # display width and height
14. print("The height of the image is: ", height)
15. print("The width of the image is: ", width)

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

Crop()

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

Image.save()

7. What module contains Pillow’s shape-drawing code ? **The 'ImageDraw' module** provides simple 2D graphics support for Image Object. Generally, we use this module to create new images, annotate or retouch existing images and to generate graphics on the fly for web use. The graphics commands support the drawing of shapes and annotation of text.

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

1. Import all required modules (numpy, imageio, scipy. ndimage, OpenCV)
2. Take Image input.
3. Check RGB value of image and convert into according to RGB values.
4. Show finale image output using cv2. imwrite()