1. What does RGBA stand for?

Ans: Red Green Blue Alpha.

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

Ans: To get the RGBA:

* Import the necessary modules
* Open the image using the ‘Image.open()’ function, providing the path to the image file:
* Convert the image to RGBA mode using the ‘covert()’ method:
* Access the pixel value if the image using the ‘load()’ method , which return a pixel access object
* Access the RGBA values of a specific pixel by providing the coordinates (x,y) of the pixel
* The ‘rgba\_value’ will be a tuples containing the RGBA values of the pixel.

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

Ans: In the context of the Pillow module, a box tuple represents a rectangular region or bounding box within an image. It is defined as a tuple containing four values: `(left, upper, right, lower)`.

each value represents in the box tuple:

- `left`: The x-coordinate of the leftmost edge of the bounding box.

- `upper`: The y-coordinate of the upper edge of the bounding box.

- `right`: The x-coordinate of the rightmost edge of the bounding box (not inclusive).

- `lower`: The y-coordinate of the lower edge of the bounding box (not inclusive).

The box tuple defines a rectangular area within the image by specifying the coordinates of the top-left and bottom-right corners. The `left` and `upper` values represent the inclusive starting point of the box, while the `right` and `lower` values represent the exclusive endpoint of the box.

The box tuple is commonly used in Pillow for various image manipulation operations, such as cropping, resizing, and extracting regions of interest from an image. It allows you to define a specific area within the image on which the operation should be performed. By specifying the appropriate values in the box tuple, you can accurately define and target specific regions of an image.

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

Ans: To find out the width and height of an Image object in a notebook using the Pillow module, you can follow these steps:

* We can use the following command:
* Import the necessary modules:
* Load the image using the `Image.open()` function, providing the path to the image file or the URL of the image:
* Get the width and height of the Image object using the `size` attribute:

Replace `"path/to/image.png"` with the actual path to the image file or use a valid image URL in place of `"https://example.com/image.png"`. The `width` and `height` variables will contain the respective dimensions of the Image object.

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

Ans: from PIL import Image

# Load the original image

image = Image.open("path/to/original\_image.png")

# Define the bounding box for the desired region

left = 0

upper = 50

right = 50

lower = 100

# Crop the image to the defined bounding box

cropped\_image = image.crop((left, upper, right, lower))

# Display the cropped image

cropped\_image.show()

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

Ans: from PIL import Image

# Load the original image

image = Image.open("path/to/original\_image.png")

# Perform modifications to the image (e.g., cropping, resizing, etc.)

# Save the modified image to a file

image.save("path/to/modified\_image.png")

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

Ans: The ‘ImageDraw’ module provides a collection of function that allows user to draw various shapes. And text, on an Image object.

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

Ans: By using the ‘ImageDraw’ module, which provides the drawing methods:

To obtain the ‘ImageDraw’ object that allows the user to perform drawing operation on an image object, user need to create it by calling the ‘ImageDraw.Draw()’ function, passing the image object as an argument.