ASSIGNMENT - 14(PYTHON)

1. What does RGBA stand for?

RGBA stands for Red, Green, Blue, and Alpha. It is a color model used to represent colors in digital displays, particularly in computer graphics and image processing.

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

In the Pillow module, we can use the `getpixel()` method to get the RGBA value of any pixel in an image.

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

A box tuple is a tuple representing a rectangular region in an image. It typically consists of four integer values defining the coordinates of the top-left corner and the bottom-right corner of the rectangle.

The box tuple is often represented as (left, top, right, bottom) or (x1, y1, x2, y2), where:

(left, top) or (x1, y1) represents the coordinates of the top-left corner of the rectangle.
(right, bottom) or (x2, y2) represents the coordinates of the bottom-right corner of the rectangle.
Here's how it works:

The left or x1 value specifies the horizontal position of the left edge of the rectangle.
The top or y1 value specifies the vertical position of the top edge of the rectangle.
The right or x2 value specifies the horizontal position of the right edge of the rectangle (exclusive).
The bottom or y2 value specifies the vertical position of the bottom edge of the rectangle (exclusive).

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

To find out the width and height of an Image object using the Pillow library in a Jupyter Notebook, you can use the `size` attribute of the Image object. Here's how we can do it:

```
from PIL import Image

# Open the image
image = Image.open("example.jpg")

# Get the width and height of the image
width, height = image.size

# Print the width and height
print("Width:", width)
print("Height:", height)
```

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

To get an Image object for a 100×100 image, excluding the lower-left quarter of it, you can use the crop() method of the Image object
Here's how we can use:

```
from PIL import Image

# Open the original image
image = Image.open("example.jpg")

# Get the width and height of the original image
width, height = image.size

# Define the box tuple for the region to crop
box = (0, height // 2, width // 2, height)

# Crop the image using the box tuple
cropped_image = image.crop(box)

# Display the cropped image
cropped_image.show()
```

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

After making changes to an Image object using the Pillow library, you can save it as an image file using the save() method of the Image object. Here's how we can do it:

```
from PIL import Image

# Open the original image
image = Image.open("example.jpg")

# Perform operations on the image (e.g., crop, resize, rotate)

# Save the modified image to a file
image.save("modified_example.jpg")
```

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

Pillow's shape-drawing code is contained in the ImageDraw module.

We can import like this:

```
from PIL import ImageDraw
```

This module provides methods to draw shapes such as lines, rectangles, ellipses, polygons, and text onto an image. It allows you to create and manipulate images with various shapes and annotations programmatically.

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

Image objects in Pillow indeed do not have drawing methods directly. Instead, you use the ImageDraw object to draw shapes, lines, and text on Image objects.

We can get an ImageDraw object by calling the ImageDraw.Draw() function and passing the Image object as an argument. Here's how we can do it:

```
from PIL import Image, ImageDraw

# Open the image

image = Image.open("example.jpg")

# Get an ImageDraw object

draw = ImageDraw.Draw(image)

# Now you can use the drawing methods of ImageDraw object
```