**1. What does RGBA stand for?**

Ans: RGBA stands for Red, Green, Blue, and Alpha. It's a colour model used to represent colours in digital images. Each pixel is defined by a combination of these four colour components: red, green, and blue for the colour itself, and alpha for the transparency or opacity of the pixel.

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

Ans: To get the RGBA value of an image using the Pillow module in Python, you can use the getpixel method. Here's an example:

from PIL import Image

# Open an image

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

# Get the RGBA value of a pixel at coordinates (x, y)

rgba\_value = image.getpixel((x, y))

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

Ans: A box tuple is a representation of a rectangular region in an image. It's defined as a tuple with four values: (left, upper, right, lower), where left and upper are the coordinates of the top-left corner of the box, and right and lower are the coordinates of the bottom-right corner of the box. It's commonly used in Pillow to define regions for cropping, pasting, and other image operations.

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

Ans: Since I cannot directly display images in this environment, I'll provide you with the code snippet to load an image in a Jupyter Notebook and find its width and height:

from PIL import Image

# Open an image

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

# Get the width and height of the image

width, height = image.size

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?**

Ans: To get an Image object for a 100×100 image excluding the lower-left quarter, you can use the crop method. Here's how you would do it:

from PIL import Image

# Open an image

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

# Define the box tuple for the region to crop (excluding lower-left quarter)

box = (0, 0, 50, 50)

# Crop the image

cropped\_image = image.crop(box)

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

Ans: To save changes made to an Image object, you can use the save method. Here's an example:

from PIL import Image

# Open an image

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

# Make changes to the image (e.g., resize, filter, etc.)

# Save the modified image

image.save("modified\_image.jpg")

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

Ans: The module that contains Pillow's shape-drawing code is ‘PIL.ImageDraw’.

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

Ans: ImageDraw objects have drawing methods in Pillow. You can create an ImageDraw object by using the ImageDraw.Draw() function and passing in an Image object. This ImageDraw object provides methods to draw shapes, lines, text, and more on the image. Here's an example:

from PIL import Image, ImageDraw

# Open an image

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

# Create an ImageDraw object

draw = ImageDraw.Draw(image)

# Use drawing methods on the ImageDraw object

draw.rectangle((x1, y1, x2, y2), outline="red")

draw.line((x1, y1, x2, y2), fill="blue")

draw.text((x, y), "Hello, Pillow!", fill="green")

# Save the modified image

image.save("drawn\_image.jpg")