1. What does RGBA stand for?

🡪RGBA stands for Red-Blue-Green-Alpha. 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?

🡪 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.  For example, the color red is represented by (255, 0, 0, 255). This color has the maximum amount of red, no green or blue, and the maximum alpha value, meaning it is fully opaque. Green is represented by (0, 255, 0, 255), and blue is (0, 0, 255, 255). White, the combination of all colors, is (255, 255, 255, 255), while black, which has no color at all, is (0, 0, 0, 255).

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

🡪Image pixels are represented with x- and y-coordinates, which respectively specify a pixel’s horizontal and vertical location in an image. The origin is the pixel at the top-left corner of the image and is specified with the notation (0, 0). The first zero represents the x-coordinate, which starts at zero at the origin and increases going from left to right. The second zero represents the y-coordinate, which starts at zero at the origin and increases going down the image. This bears repeating: y-coordinates increase going downward.

Many of Pillow’s functions and methods take a box tuple argument. This means Pillow is expecting a tuple of four integer coordinates that represent a rectangular region in an image. The four integers are, in order, as follows:

* Left: The x-coordinate of the leftmost edge of the box.
* Top: The y-coordinate of the top edge of the box.
* Right: The x-coordinate of one pixel to the right of the rightmost edge of the box. This integer must be greater than the left integer.
* Bottom: The y-coordinate of one pixel lower than the bottom edge of the box. This integer must be greater than the top integer.

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

🡪 imageObj.size is a tuple of two integers, which can give us the width and the height of an image object.

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

🡪 Excluding the lower-left quarter of a 100x100 image, we can use the following code-

from PIL import Image, ImageDraw

im = Image.open(r'C:\Users\user\Desktop\DS\Sankalp pic.jpg')

draw = ImageDraw.Draw(im)

draw.rectangle((0,50,50,100),fill=’black’)

im.show()

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

🡪 Call the imageObj.save('new\_filename.png') method of the Image object.

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

🡪 Pillow’s ImageDraw module has shape-drawing code.

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.