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

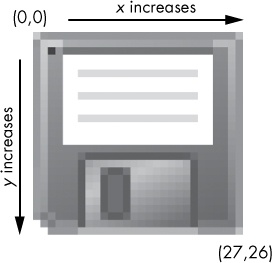
**Ans:** An RGBA value is a group of numbers that specify the amount of red, green, blue, and alpha (or transparency) in a color.

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

**Ans:** Pillow offers the ImageColor.getcolor() function so we don’t have to memorize RGBA values for the colors you want to use. This function takes a color name string as its first argument, and the string 'RGBA' as its second argument, and it returns an RGBA tuple.

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

**Ans:** 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:



**The x- and y-coordinates of a 27×26 image of some sort of ancient data storage device**

* *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?**

**Ans**: # import required module

from PIL import Image

# get image

filepath = "vikrant.png"

img = Image.open(filepath)

# get width and height

width = img.width

height = img.height

# display width and height

print("The height of the image is: ", height)

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?**

**Ans:** from PIL import Image

# Load the original image

original\_image = Image.open("original\_image.jpg")

# Create a new image with the same size as the original

new\_image = Image.new("RGB", (100, 100))

# Crop the original image to exclude the lower-left quarter

cropped\_image = original\_image.crop((0, 0, 75, 75)) # Exclude the region from (0, 0) to (75, 75)

# Paste the cropped image onto the new image, starting from the top-left corner

new\_image.paste(cropped\_image, (0, 0))

# Optional: Save the new image

new\_image.save("new\_image.jpg")

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

**Ans:** To save the Image object to an image file, we call the save() method.

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

**Ans:** ImageDraw module contains Pillow’s shape-drawing code.

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

**Ans:** Python’s Pillow which is a fork of the discontinued Python Imaging Library (PIL) is a powerful library that is capable of adding image processing capabilities to your python code. Pillow offers many modules that ease the process of working and modifying images.

In this article, we will have a look at the ImageDraw module of this library. ImageDraw provides a variety of methods to, as its name suggests, draw on images. With the help of this module, we can draw lines, circles, rectangles and, even write and format text on an image.

**Drawing common shapes on image**

**The image we will be using can be displayed using PIL as follows:**

# Importing Image and ImageDraw from PIL

from PIL import Image, ImageDraw

# Opening the image to

# be used and displaying it

img = Image.open('img\_path.png')

img.show()