1. What does RGBA stand for?

Answer=

* RGBA stands for red, green, blue, and alpha (transparency) in the color
* 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?

Answer=

You can use the getpixel() method of the Image class from the PIL (Python Imaging Library) module, which is a fork of the original PIL library and is also known as Pillow. The getpixel() method takes a tuple of x and y coordinates and returns a tuple of the RGBA values of the pixel at that position.

from PIL import Image  
im = Image.open('image.jpg') # Open an image  
# Get the RGBA value of a pixel at position (x, y)  
x, y = (100, 100)  
rgba = im.getpixel((x, y))  
print(rgba)

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

Answer=

A box tuple is a tuple value of four integers: the left edge x-coordinate, the top edge y-coordinate, the width, and the height, respectively

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

Answer=

from google.colab import files

uploaded = files.upload()

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving shiva.jpg to shiva.jpg

from PIL import Image

myImg = Image.open('shiva.jpg')

w,h = myImg.size

w,h

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

Answer=

ImageObject.crop((0, 50, 50, 50))

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

Answer=

By Calling the imageObj.save('new\_filename.png') method of the Image object.

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

Answer=

The ImageDraw module contains code to draw on images.

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

Answer=

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.