**Assignment\_14**

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

**Ans:**

RGBA stands for Red Green Blue Alpha. It is a color model used to represent colors in digital images and computer graphics. The first three letters (RGB) represent the three primary colors of light, red, green, and blue, and the fourth letter (A) represents the alpha channel, which controls the transparency of the color.

1. **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 of the `Image` object. Here's an example code snippet:

from PIL import Image

img = Image.open('image.png')

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

rgba = img.getpixel((x, y))

# Print the RGBA value

print(rgba)

In the above code, ‘image.png’ is the name of the image file that you want to open. The ‘getpixel()’ method takes the coordinates of the pixel whose RGBA value you want to retrieve as a tuple ‘(x, y)’. The method returns a tuple containing the RGBA values of the pixel at the given coordinates.

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

**Ans:**

A box tuple is defined as a tuple with four integer values in the. Here, left and upper represent the coordinates of the top-left corner of the rectangular region, while right and lower represent the coordinates of the bottom-right corner of the rectangular region. The left and upper values are inclusive, meaning that the pixel at (left, upper) is included in the box, while the right and lower` values are exclusive, meaning that the pixel at (right, lower) is not included in the box.

When you pass a box tuple to a method or function in the Pillow module, it uses the tuple to define a rectangular region in the image. For example, you can use the ‘crop’ method of the ‘Image’ object to extract a rectangular region of an image defined by a box tuple.

from PIL import Image

# Open the image

img = Image.open('image.png')

# Define the box tuple for the region of interest

box = (100, 100, 300, 300)

# Crop the image to the region of interest

cropped\_img = img.crop(box)

# Save the cropped image

cropped\_img.save('cropped\_image.png')

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

**Ans:**

Example for how can we load an image file into a Jupyter notebook using the Pillow module and find out its width and height:

from PIL import Image

# Open the image file

img = Image.open('my\_image.jpg')

# Get the width and height of the image

width, height = img.size

# Print the width and height

print(f"The image has a width of {width} pixels and a height of {height} pixels.")

1. **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 100x100 image, excluding the lower-left quarter of it, you can use the `crop()` method of the `Image` object in combination with a box tuple.

Ex:

from PIL import Image

img = Image.open('my\_image.jpg')

box = (50, 0, 100, 50)

cropped\_img = img.crop(box)

cropped\_img.show()

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

**Ans:**

To save an `Image` object as an image file after making changes to it, you can use the `save()` method of the `Image` object. The `save()` method takes a file path as an argument and writes the image to the specified file path.

Ex:

from PIL import Image

img = Image.open('my\_image.jpg')

# Save the modified image to a file

img.save('modified\_image.jpg')

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

**Ans:**

Pillow's shape-drawing code is located in the `ImageDraw` module. The `ImageDraw` module provides a way to draw 2D graphics on `Image` objects using various shapes such as lines, rectangles, circles, arcs, polygons, and text. It also provides methods for filling shapes with solid colors, gradients, and patterns.

Ex:

from PIL import Image, ImageDraw

# Open the image file

img = Image.open('my\_image.jpg')

# Create an ImageDraw object

draw = ImageDraw.Draw(img)

# Draw a red circle

draw.ellipse((50, 50, 150, 150), fill='red')

# Save the modified image

img.save('modified\_image.jpg')

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

**Ans:**

The `ImageDraw` module provides a way to draw shapes and text on `Image` objects in Pillow. `ImageDraw` provides a `Draw` object that has methods for drawing various shapes and text on an `Image`.

To get an `ImageDraw` object, you first need to create one by calling the `ImageDraw.Draw()` method and passing in an `Image` object as an argument. For example:

from PIL import Image, ImageDraw

img = Image.open('my\_image.jpg')

# Create an ImageDraw object

draw = ImageDraw.Draw(img)

# Draw a line on the image

draw.line((0, 0, 100, 100), fill='red')

# Save the modified image

img.save('modified\_image.jpg')