Assignment_14

1. What does RGBA stand for?

> RGBA stands for Red, Green, Blue, and Alpha. RGBA colors are represented by four values:

```
Red: The amount of red in the color.
Green: The amount of green in the color.
Blue: The amount of blue in the color.
Alpha: The opacity of the color.
```

> The alpha value is a number between 0 and 1. A value of 0 means that the color is completely transparent, and a value of 1 means that the color is completely opaque.
>
> - `rgba(255, 0, 0, 0.5)` This color is red with an alpha value of 0.5, which means that it is semi-transparent.

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

> To get the RGBA value of any images from the Pillow module, we can use the getdata() method. The getdata() method returns a sequence of (red, green, blue, alpha) tuples, where each tuple represents the color of a pixel in the image.```

```
import PIL.Image

image = PIL.Image.open('image.png')
rgba_values = image.getdata()

for red, green, blue, alpha in rgba_values:
    print(red, green, blue, alpha)
```

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

> A box tuple, also known as a tuple struct, is a data structure in Python that is similar to a tuple, but the values to be stored in a box tuple can be mutable objects, such as lists or dictionaries.
> It works as:

```
def my_function(box_tuple):
    box_tuple[0].append(4)
    box_tuple[1]['key'] = 'new value'

box_tuple = ([1, 2, 3], {'key': 'value'})
my_function(box_tuple)
print(box_tuple)
```

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

To find out the width and height of an Image object, you can use the **size** property.
lets find the width and height of image.

```
import PIL.Image

image = PIL.Image.open('pravash.png')
width = image.size[0]
height = image.size[1]
print(width, height)
```

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

To get an Image object for a 100×100 image, we can use the crop() method. The crop() method takes four arguments: the top left corner coordinates, and the width and height of the crop.
let get an Image object for a 100×100 image, excluding the lower-left quarter of it:

```
import PIL.Image

image = PIL.Image.open('pravash.png')
width = image.size[0]
height = image.size[1]

crop_coordinates = (0, 0, width / 2, height / 2)
cropped_image = image.crop(crop_coordinates)
print(cropped_image)
```

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

> To save an Image object as an image file, we can use the save() method. The save() method takes two arguments: the filename of the image file, and the format of the image file.

> syntax: `image.save('cropped_image.png', 'PNG')`

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

> The module that contains Pillow’s shape-drawing code is the **ImageDraw** module. The ImageDraw module provides a number of functions that can be used to draw shapes on images. These shapes include lines, rectangles, circles, and ellipses.

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

> Image objects do not have drawing methods because they are **immutable**. This means that they cannot be changed once they are created. To draw on an image, you need to create a **Draw** object by using the **Draw()** method on an Image object.