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

Solution - RGBA(Red-Green-Blue-Alpha).The RGB color model is extended in this specification to include “alpha” to allow specification of the opacity of a color.

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

Solution - Pillow offers the ImageColor. getcolor() function so you 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?**

Solution - The box. tuple submodule provides read-only access for the tuple userdata type. It allows, for a single tuple: selective retrieval of the field contents, retrieval of information about size, iteration over all the fields, and conversion to a Lua table.

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

Solution - PIL.Image.open() is used to open the image and then .width and .height property of Image are used to get the height and width of the image. The same results can be obtained by using .size property.

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

Solution - To get an image object for a 100x100 image, excluding the lower left quarter of it in Python, you can use the Pillow library. Here is an example code snippet that demonstrates this:

```
from PIL import Image

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

cropped_img = img.crop((0, 0, 75, 75))

resized_img = cropped_img.resize((100, 100))

resized_img.show()
```

In this code, `Image.open()` is used to load the image file, and `Image.crop()` is used to crop the image to exclude the lower left quarter. The `crop()` method takes a tuple of four values representing the left, upper, right, and lower pixel coordinates of the crop box. In this case, we set the crop box to `(0, 0, 75, 75)` to exclude the lower left quarter of the original image. Then, `Image.resize()` is used to resize the cropped image to 100x100. Finally, `Image.show()` is used to display the resulting image.

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

Solution - To save images, we can use the PIL.save() function. This function is used to export an image to an external file. But to use this function, first, we should have an object which contains an image.

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

Solution - The ImageDraw module provides simple 2D graphics for Image objects. You can use this module to create new images, annotate or retouch existing images, and to generate graphics on the fly for web use.

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

Solution - In Python, the `Pillow` library provides an `ImageDraw` module that contains a set of drawing primitives that can be used to draw on `Image` objects. To use the drawing methods, you need to create an `ImageDraw` object from the `Image` object.

```
from PIL import Image, ImageDraw

img = Image.new('RGB', (200, 200), color='white')

draw = ImageDraw.Draw(img)

draw.rectangle((50, 50, 150, 150), outline='red')

img.save('output.png')
```

In this code, `Image.new()` is used to create a new `Image` object with a white background and dimensions of 200x200 pixels. Then, an `ImageDraw` object is created from the `Image` object using `ImageDraw.Draw()`. Finally, the `rectangle()` method of the `ImageDraw` object is called to draw a red rectangle on the image. The `rectangle()` method takes a tuple of four values representing the left, upper, right, and lower coordinates of the rectangle. The resulting image is then saved to a file using the `save()` method of the `Image` object.