How to Use PIL for Working With Images

As we've mentioned, for the project in this module, you'll use the Python Imaging Library to process a bunch of images. So, how does that work?

When using PIL, we typically create **Image** objects that hold the data associated with the images that we want to process. On these objects, we operate by calling different methods that either return a new image object or modify the data in the image, and then end up saving the result in a different file.

For example, if we wanted to resize an image and save the new image with a new name, we could do it with:

```
1   from PIL import Image
2   im = Image.open("example.jpg")
3   new_im = im.resize((640,480))
4   new_im.save("example_resized.jpg")
5
```

In this case, we're using the resize method that returns a new image with the new size, and then we save it into a different file. Or, if we want to rotate an image, we can use code like this:

```
1   from PIL import Image
2   im = Image.open("example.jpg")
3   new_im = im.rotate(90)
4   new_im.save("example_rotated.jpg")
5
```

This method also returns a new image that we can then use to create the new rotated file. Because the methods return a new object, we can even combine these operations into just one line that rotates, resizes, and saves:

```
1   from PIL import Image
2   im = Image.open("example.jpg")
3   im.rotate(180).resize((640,480)).save("flipped_and_resized.jpg")
4
```

There's a ton more that you can do with the PIL library. Have a look at the docs and try it on your computer!