PYTHON IS AWESOME:

PILLOW(PIL) LIBRARY

By: Rykir Evans
& Vicky Heredia

PIL/PILLOW LIBRARY OVERVIEW

What is Pillow?

- Pillow is a Python imaging library used for opening, editing, and saving images.
- It's widely used in basic graphics work and simple image manipulation tasks.

Why is it useful?

- Great for beginners
- Allows visual /creative projects
- Direct access to individual pixels
- Useful for automation (like watermarking or resizing many images)

What does Pillow do?

- Open images in many formats (PNG, JPG, etc.)
- Resize, crop, and rotate images
- Add filters such as blur, sharpen, or grayscale
- Draw shapes/text on images
- Convert between image formats
- Save edited images
- Brightness adjustment

CODE EXAMPLE #1: RESIZE, GRAYSCALE AND ADD TEXT

4143-Java-Python > Assignments > PythonIsAwesome > main.py > ...

```python
from PIL import Image, ImageDraw, ImageFont, ImageEnhance

# Open image
img = Image.open("my_photo.jpg")
draw = ImageDraw.Draw(img)

# Load a font
font = ImageFont.truetype("/usr/share/fonts/truetype/dejavu/DejaVuSans.ttf", 50)

# Draw text
draw.text((100, 50), "Hello!", fill="blue", font=font)

# Convert to grayscale, there are 2 ways of
# converting to grayscale

# Option 1:
enhanceImg = ImageEnhance.Color(img)
saturatedImg = enhanceImg.enhance(0)

#Option 2:
# gray_img = img.convert("L")
# small_gray = gray_img.resize((300, 300))

# Resize image
small_gray = saturatedImg.resize((300, 300))

# Show edited image
small_gray.show()

# Save edited image
small_gray.save("my_photo_gray_small.jpg")
```

IMAGE OUTPUT

Original

my_photo.jpg U my_photo_gray_small.jpg U

~/projects/4143-Java-Python/Assignments/PythonIsAwesome/my_photo.jpg • Untracked

Output

my_photo.jpg U my_photo_gray_small.jpg U

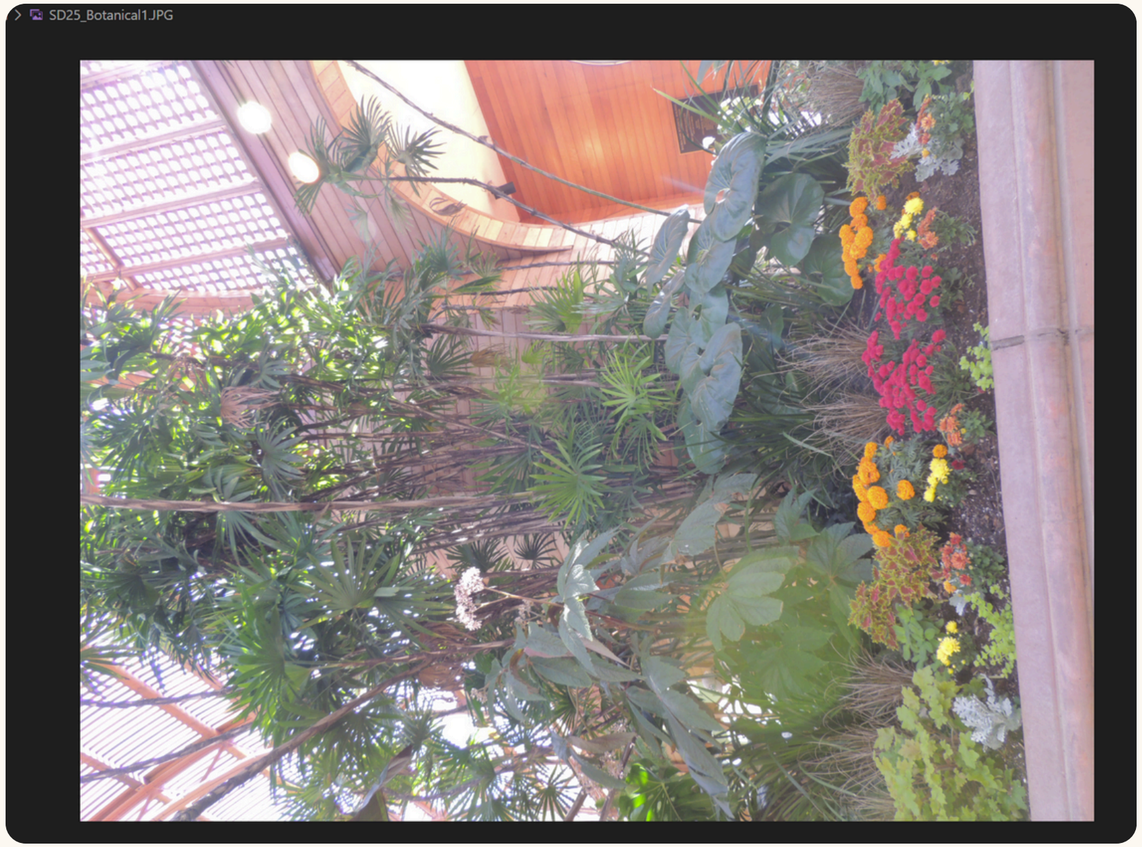
4143-Java-Python > Assignments > PythonIsAwesome > my_photo_gray_small.jpg

CODE EXAMPLE #2: ROTATION

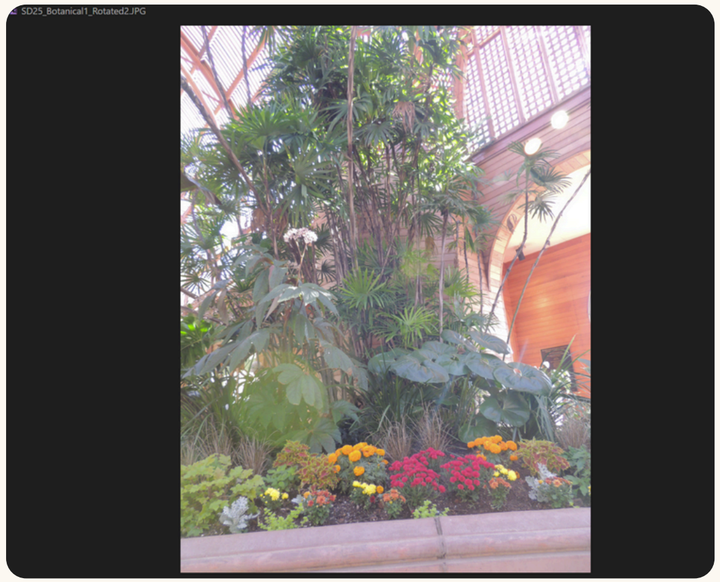
```python
from PIL import Image, ImageEnhance, ImageOps

img = Image.open("SD25_Botanical1.JPG")

################################
## Orientation Manipulation ##
################################

# Simple rotation (prone to data loss)
rotated = img.rotate(270)

rotated.save("SD25_Botanical1_Rotated1.JPG")

# Transpose rotation (produces correct results)

rotated = img.transpose(Image.ROTATE_270)

rotated.save("SD25_Botanical1_Rotated2.JPG")
```

IMAGE OUTPUT

Original



Output



CODE EXAMPLE #3: BRIGHTNESS

```
56  ##############################
57  ## Brightness Manipulation ##
58  ##############################
59
60  # Brightness enhancement object
61  img = Image.open("SD25_Botanical1_Rotated2.JPG")
62  br_enhancer = ImageEnhance.Brightness(img)
63
64  # Convert brightness to 65%
65  dimImage = br_enhancer.enhance(0.65)
66  dimImage.save("SD25_Botanical1_Dim.JPG")
67
68  # Convert brightness to 135%
69  brightImage = br_enhancer.enhance(1.35)
70  brightImage.save("SD25_Botanical1_Bright.JPG")
71
```

IMAGE OUTPUT

Original

Output (Dim)

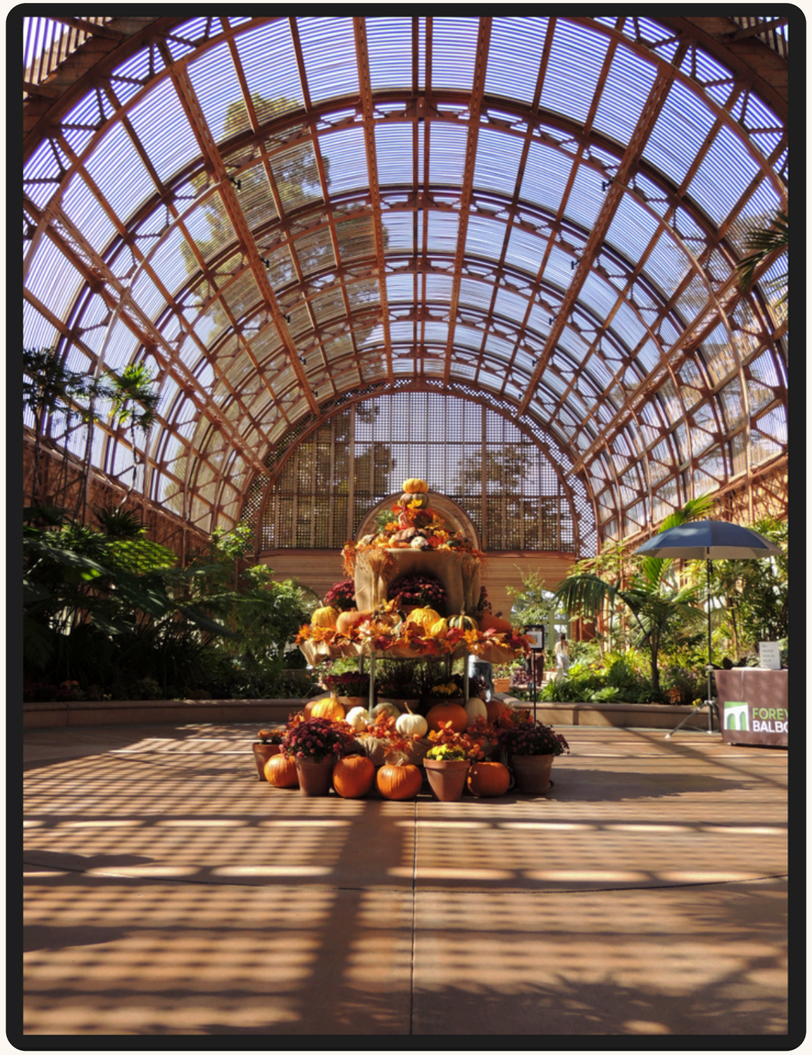
Output (Brighter)

CODE EXAMPLE #3: CONTRAST

```
72  ############################
73  ## Contrast Manipulation ##
74  ############################
75  img = Image.open("SD25_Botanical2.JPG")
76
77  # Needed because original stored image was externally rotated
78  # EXIF = Exchangeable Image File Format
79  # Pillow doesn't automatically read, this command sets it right
80  img = ImageOps.exif_transpose(img)
81
82  con_enhancer = ImageEnhance.Contrast(img)
83
84  # Increase contrast (1.0 = original)
85  higher_contrast = con_enhancer.enhance(1.2)   # 20% more contrast
86  higher_contrast.save("SD25_Botanical2_HighCon.JPG")
87
88  # Decrease contrast
89  lower_contrast = con_enhancer.enhance(0.8)    # 20% less contrast
90  lower_contrast.save("SD25_Botanical2_LowCon.JPG")
91
```

IMAGE OUTPUT

Original



Output (More)

Output (Less)

CODE EXAMPLE #4: FILTER, SATURATION AND CROP THE IMG

4143-Java-Python > Assignments > PythonIsAwesome > main2.py > ...

```python
from PIL import Image, ImageFilter, ImageEnhance

img = Image.open("sunsetSanDiego.jpg")

# Blur the image:
# when radius=1 (slight blur), when radius=10 (very blurry)
blurred = img.filter(ImageFilter.GaussianBlur(radius=5))

# Increase saturation:
# 1.0 = normal, 2.0 = very saturated, 0 = grayscale
enhancer = ImageEnhance.Color(blurred)
saturated = enhancer.enhance(4)


# Crop the image
# Crop box = (left, top, right, bottom)
cropped = saturated.crop((50, 50, 1500, 900))

cropped.show()
cropped.save("sunset_blur_sat_crop.jpg")

```

IMAGE OUTPUT

Original

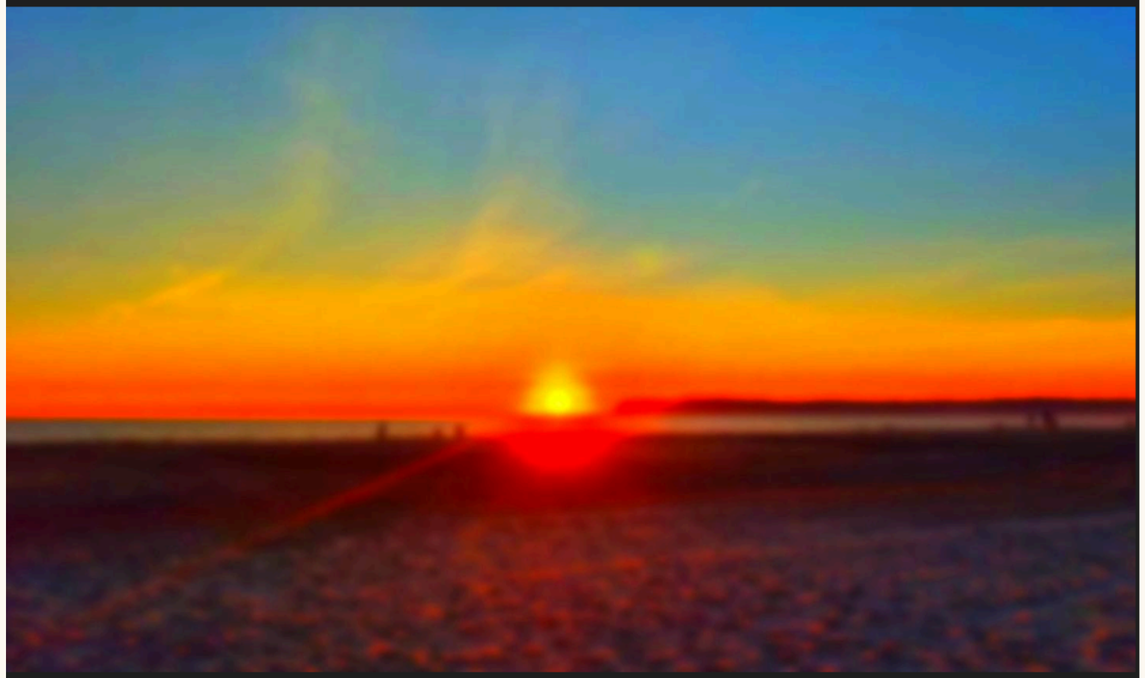
my_photo.jpg U | my_photo_gray_small.jpg U | sunsetSanDiego.jpg U

4143-Java-Python > Assignments > PythonIsAwesome > sunsetSanDiego.jpg

Output

my_photo_gray_small.jpg U | sunsetSanDiego.jpg U | sunset_blur_sat_crop.jpg U

4143-Java-Python > Assignments > PythonIsAwesome > sunset_blur_sat_crop.jpg



NOTES

Extreme Options

- Our exhibited use case is surface-level
- Several functions for pixel-level manipulation or high-level editing option

Confusing Return Values

- Some functions return a copy with edits, while others manipulate the image directly in memory
- Ensure you know which is which
- i.e. Enhancers return copies, ImageDraw acts directly

No EXIF

- Pillow does not include exchangeable Image File Format metadata
- This includes rotations applied previously, camera info, shooting settings, etc.

THANK YOU!