Mastering Python
الدرس # 2_9
معالجة الصور Pillow

By:

Hussam Hourani

V1.0 -NOV 2019

Agenda

- What is Pillow
- examples

what Pillow?

Pillow : The Python Imaging Library adds image processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

http://pillow.readthedocs.io

Supported Formats

Fully supported formats

BMP
DIB
EPS
GIF
Reading sequences
Saving
Reading local images
ICNS
ICO
IM
JPEG
JPEG 2000
MSP
PCX
PNG
PPM
SGI
SPIDER
Writing files in SPIDER format
TGA
TIFF
Reading Multi-frame TIFF Images
Saving Tiff Images
WebP
Saving sequences
XBM

Read-only formats

BLP
CUR
DCX
DDS
FLI, FLC
FPX
FTEX
GBR
GD
IMT
IPTC/NAA
MCIDAS
MIC
MPO
PCD
PIXAR
PSD
WAL
XPM

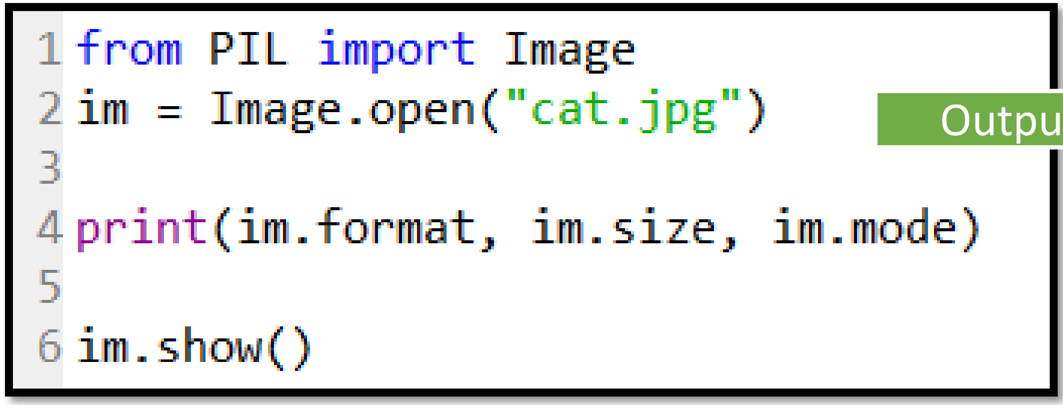
Write-only formats

PALM
PDF
XV Thumbnails

Identify-only formats

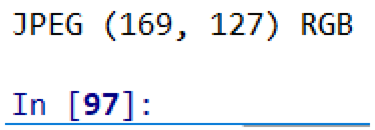
BUFR
FITS
GRIB
HDF5
MPEG
WMF

Loading, showing & Blurring an image



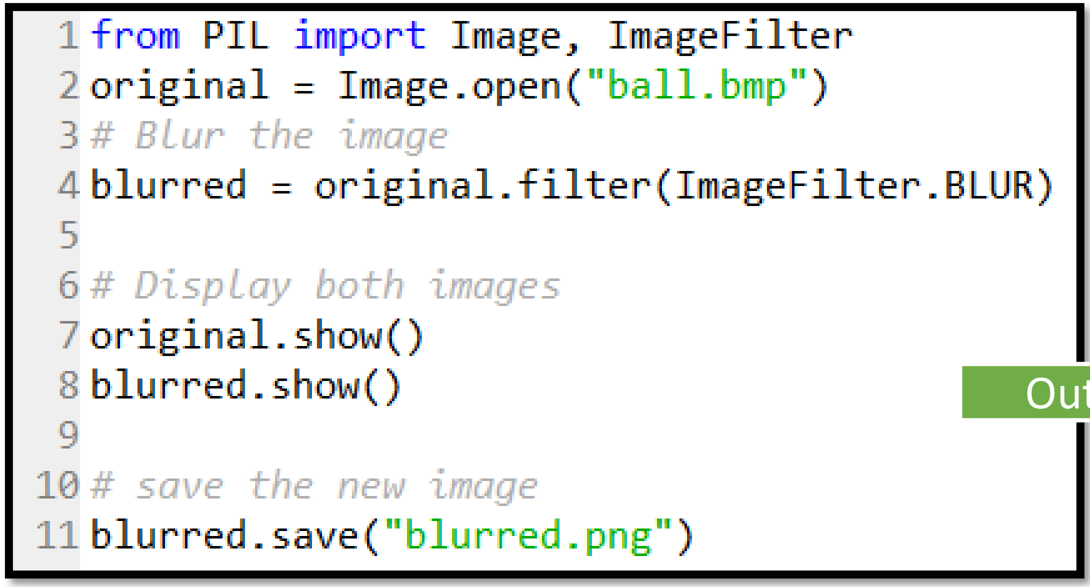
```python
from PIL import Image
im = Image.open("cat.jpg")

print(im.format, im.size, im.mode)

im.show()
```

Output



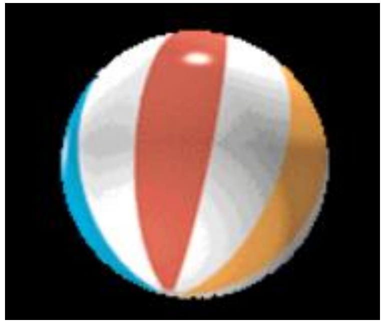
```
JPEG (169, 127) RGB

In [97]:
```

Note Common "modes" are:
"L" for greyscale images,
"RGB" for true color images,
"CMYK" for pre-press images.

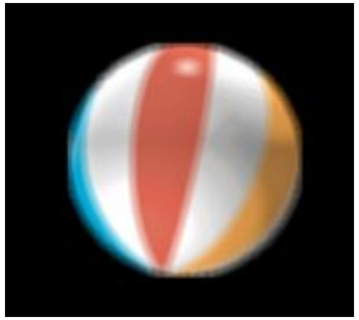


```python
from PIL import Image, ImageFilter
original = Image.open("ball.bmp")
# Blur the image
blurred = original.filter(ImageFilter.BLUR)

# Display both images
original.show()
blurred.show()

# save the new image
blurred.save("blurred.png")
```

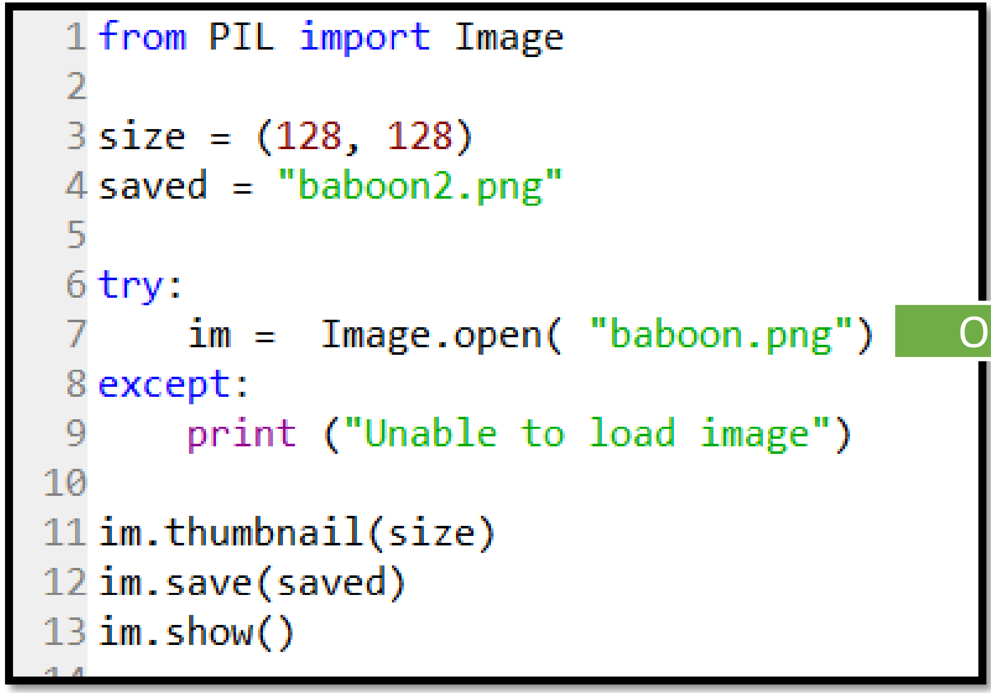
Output



Original Image

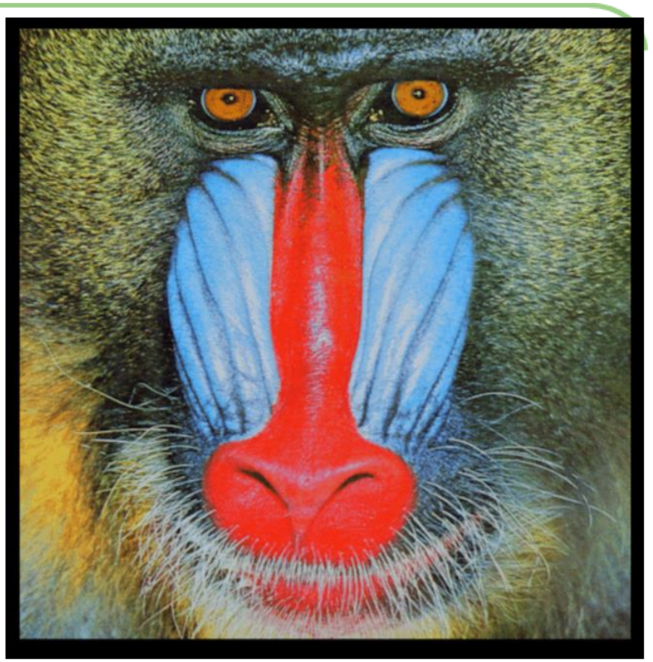


Creating Thumbnails

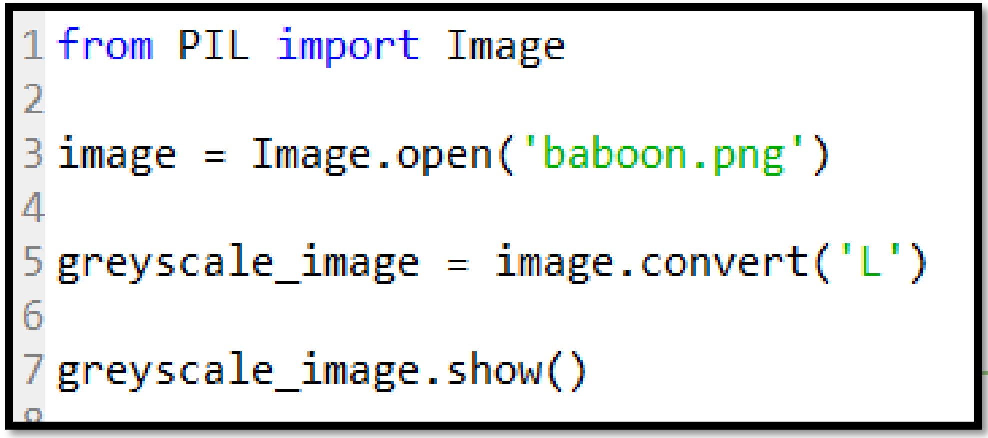


```python
from PIL import Image

size = (128, 128)
saved = "baboon2.png"

try:
    im =  Image.open( "baboon.png")
except:
    print ("Unable to load image")

im.thumbnail(size)
im.save(saved)
im.show()
```

Output

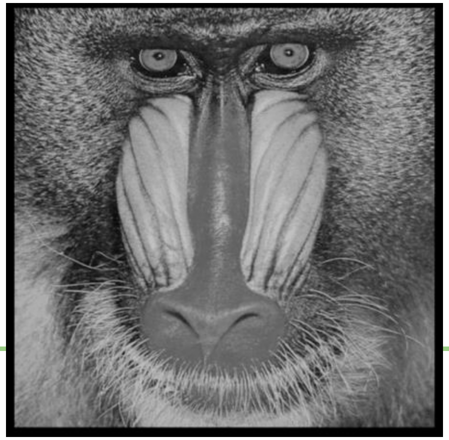


Original Image

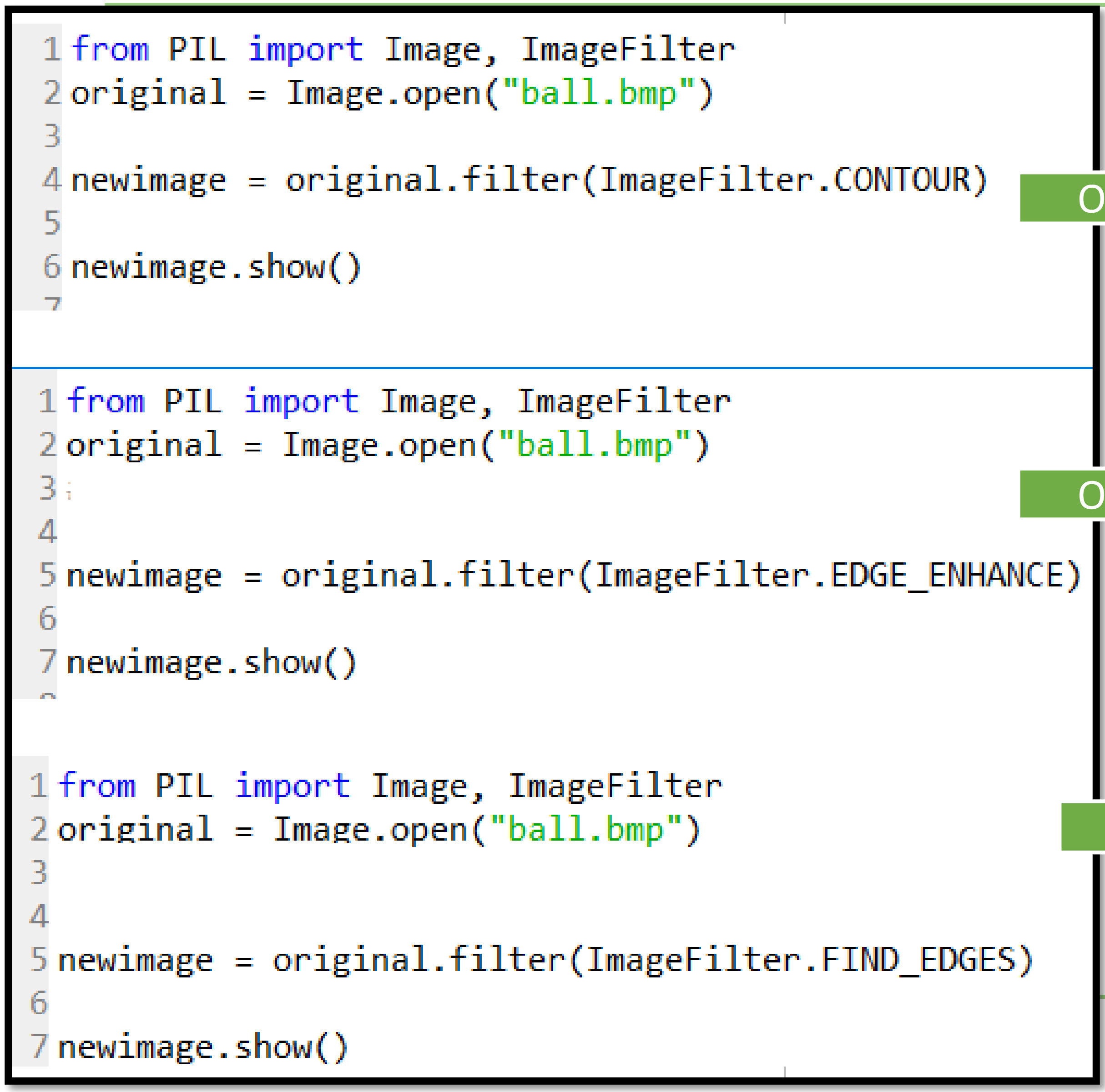


```python
from PIL import Image

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

greyscale_image = image.convert('L')

greyscale_image.show()
```

Output



Filters

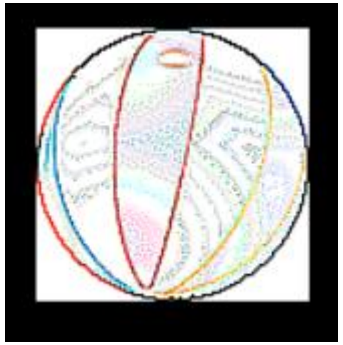


```
from PIL import Image, ImageFilter
original = Image.open("ball.bmp")

newimage = original.filter(ImageFilter.CONTOUR)

newimage.show()
```

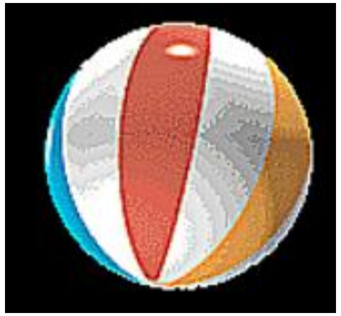
```
from PIL import Image, ImageFilter
original = Image.open("ball.bmp")


newimage = original.filter(ImageFilter.EDGE_ENHANCE)

newimage.show()
```

```
from PIL import Image, ImageFilter
original = Image.open("ball.bmp")


newimage = original.filter(ImageFilter.FIND_EDGES)

newimage.show()
```

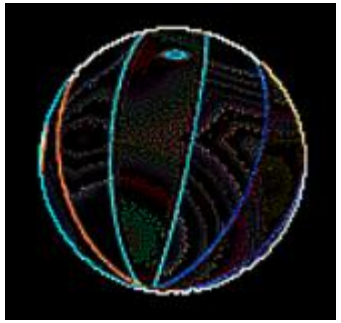
Output



Output



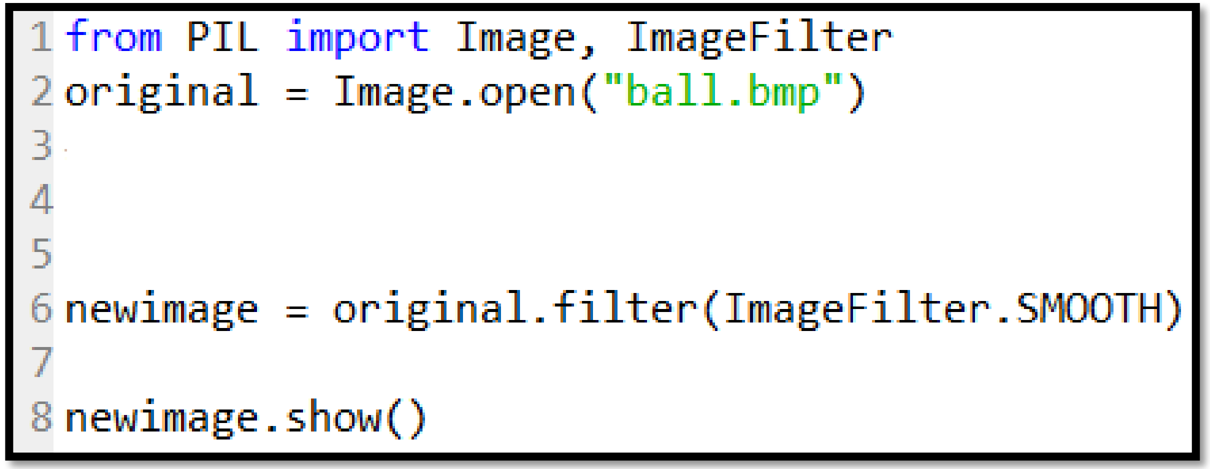
Output



Original Image

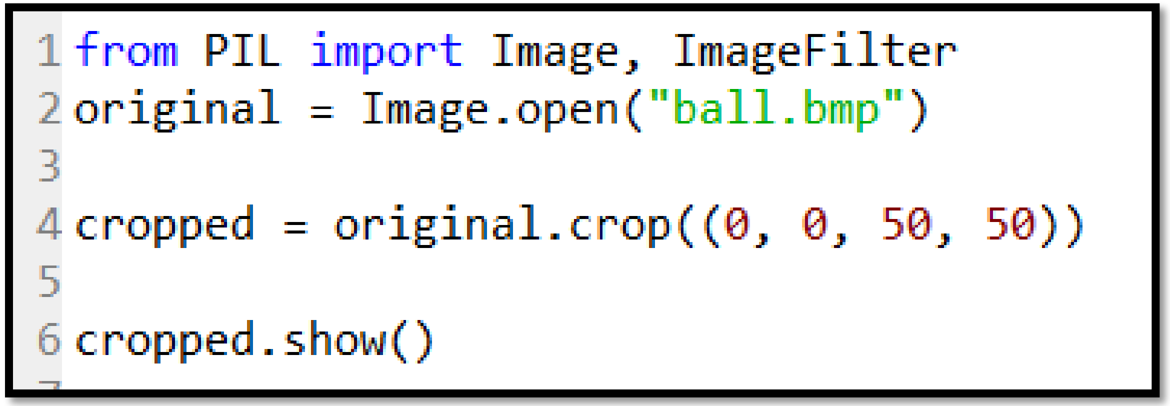
filters:
BLUR
CONTOUR
DETAIL
EDGE_ENHANCE
EDGE_ENHANCE_MORE
EMBOSS
FIND_EDGES
SMOOTH
SMOOTH_MORE
SHARPEN

Filter & Crop



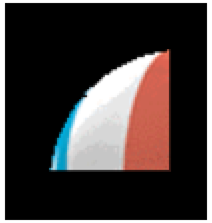
```python
from PIL import Image, ImageFilter
original = Image.open("ball.bmp")



newimage = original.filter(ImageFilter.SMOOTH)

newimage.show()
```

Output

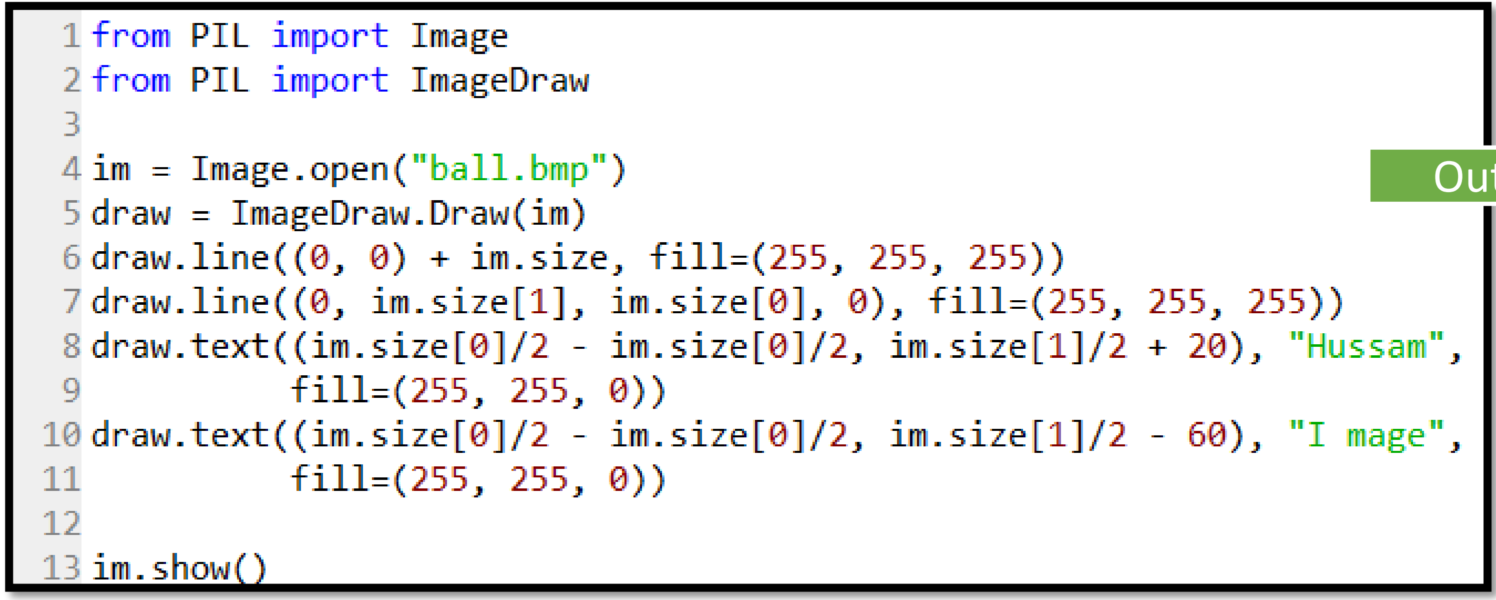


```python
from PIL import Image, ImageFilter
original = Image.open("ball.bmp")

cropped = original.crop((0, 0, 50, 50))

cropped.show()
```

Output

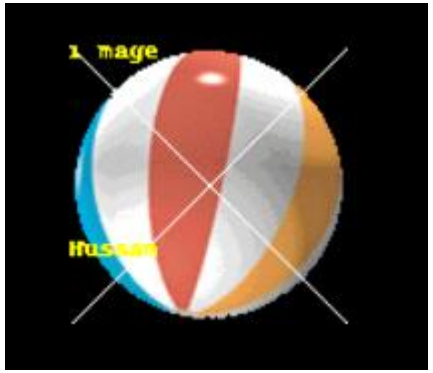


More functions

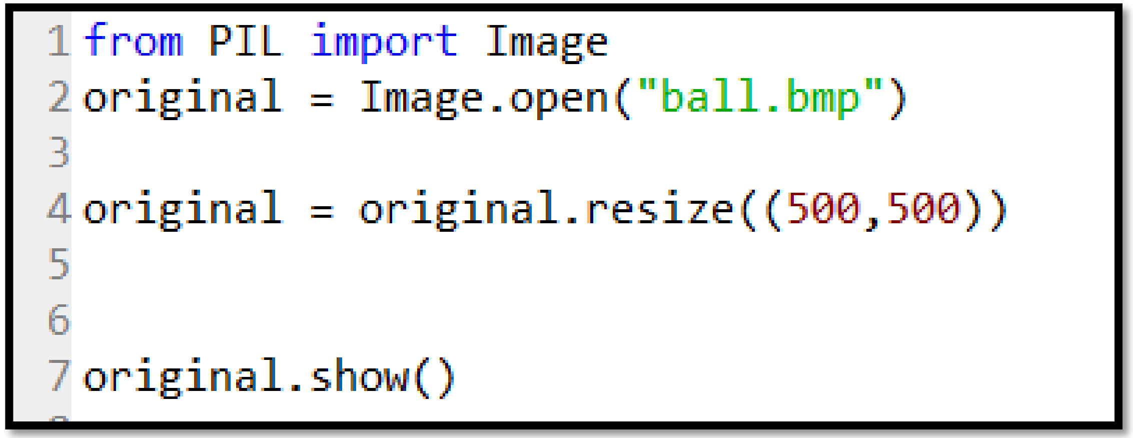


```python
from PIL import Image
from PIL import ImageDraw

im = Image.open("ball.bmp")
draw = ImageDraw.Draw(im)
draw.line((0, 0) + im.size, fill=(255, 255, 255))
draw.line((0, im.size[1], im.size[0], 0), fill=(255, 255, 255))
draw.text((im.size[0]/2 - im.size[0]/2, im.size[1]/2 + 20), "Hussam",
          fill=(255, 255, 0))
draw.text((im.size[0]/2 - im.size[0]/2, im.size[1]/2 - 60), "I mage",
          fill=(255, 255, 0))

im.show()
```

Output

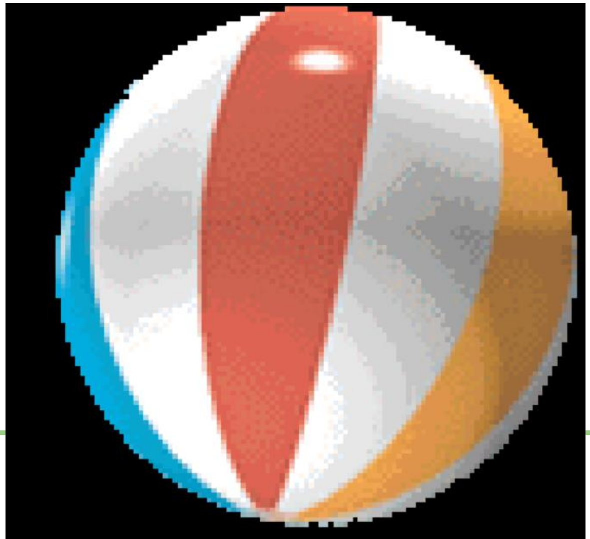


(0, 0) in the upper left corner.

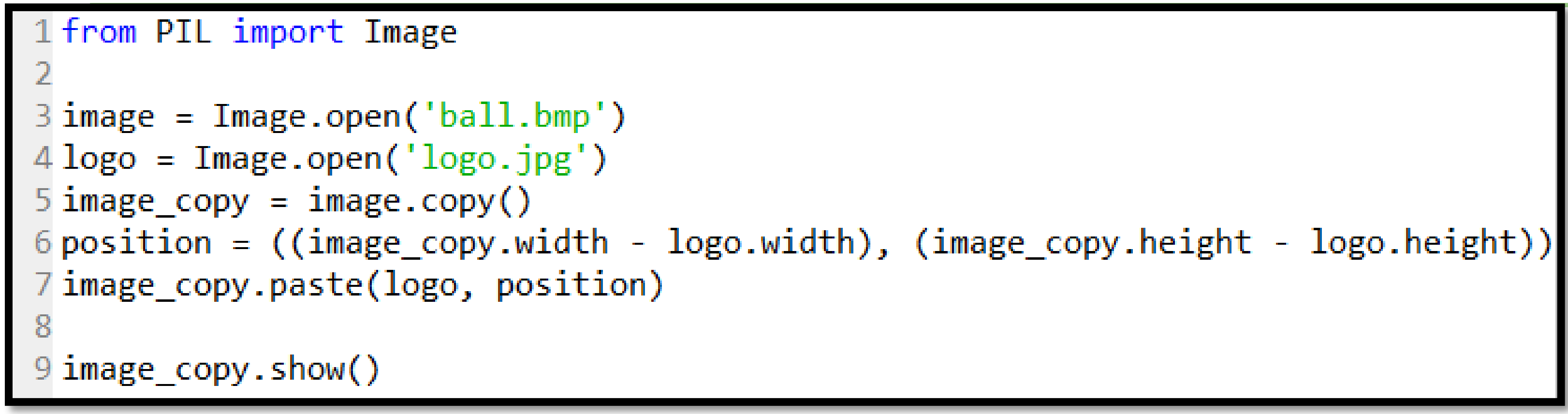


```python
from PIL import Image
original = Image.open("ball.bmp")

original = original.resize((500,500))


original.show()
```

Output

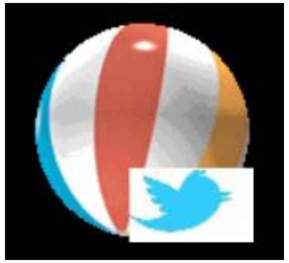


More functions

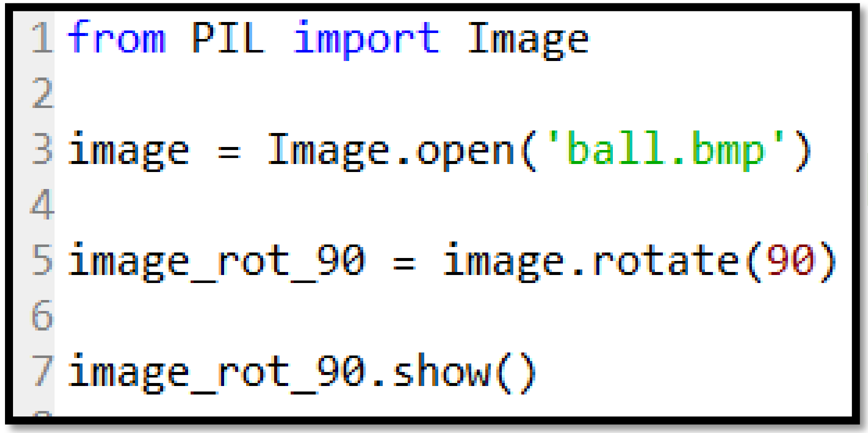


```python
from PIL import Image

image = Image.open('ball.bmp')
logo = Image.open('logo.jpg')
image_copy = image.copy()
position = ((image_copy.width - logo.width), (image_copy.height - logo.height))
image_copy.paste(logo, position)

image_copy.show()
```

Output

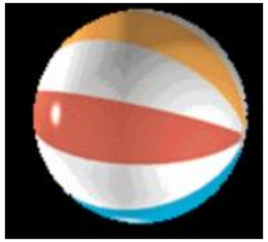


```
flower = Image.open(‘first Image')
butterfly = Image.open(‘second image')
flower.paste(butterfly, (40, 10))
```



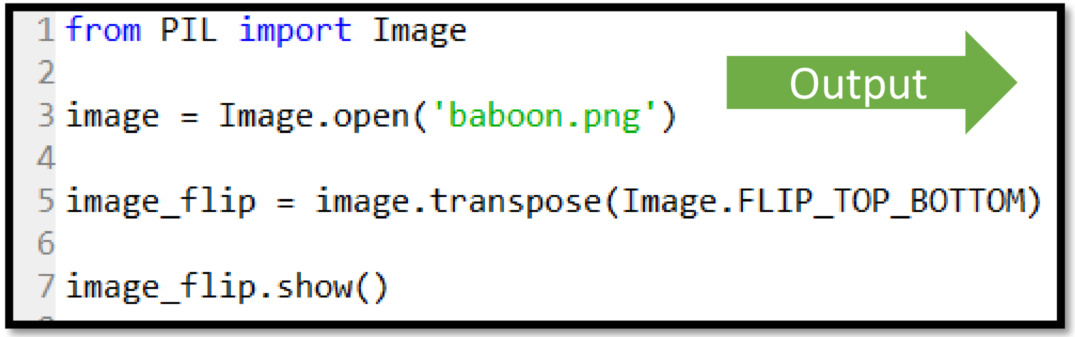
```python
from PIL import Image

image = Image.open('ball.bmp')

image_rot_90 = image.rotate(90)

image_rot_90.show()
```

Output



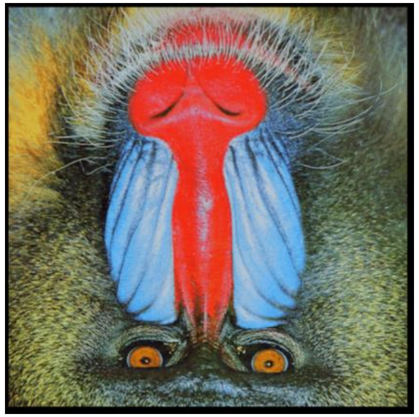
http://adventuresinmachinelearning.com/keras-tutorial-cnn-11-lines/

More functions

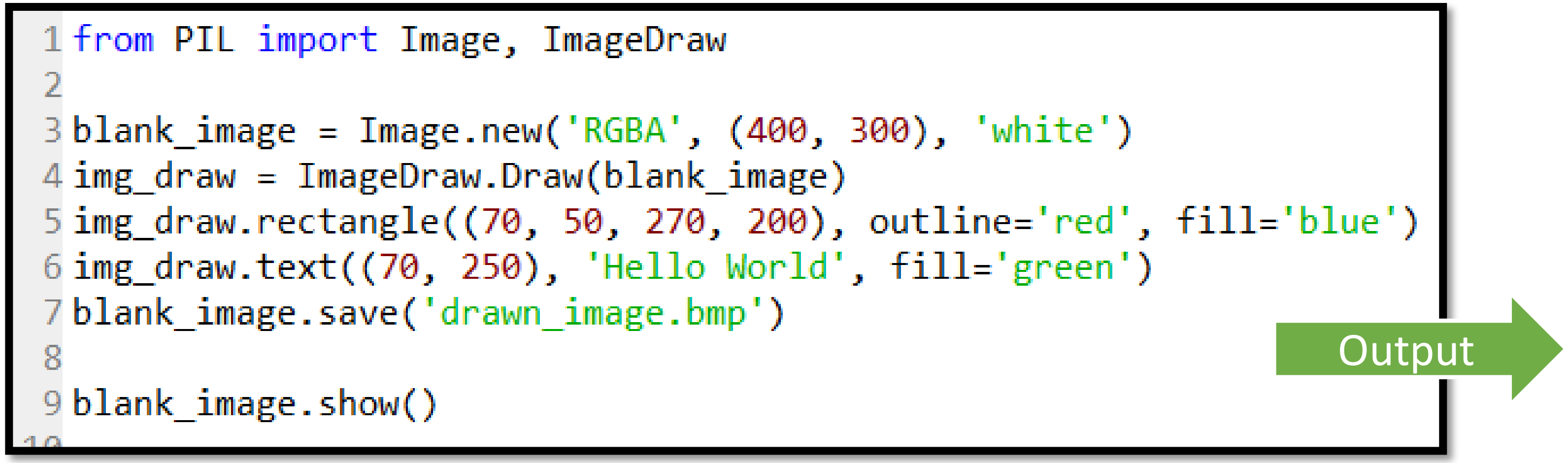


```python
from PIL import Image

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

image_flip = image.transpose(Image.FLIP_TOP_BOTTOM)

image_flip.show()
```

Output

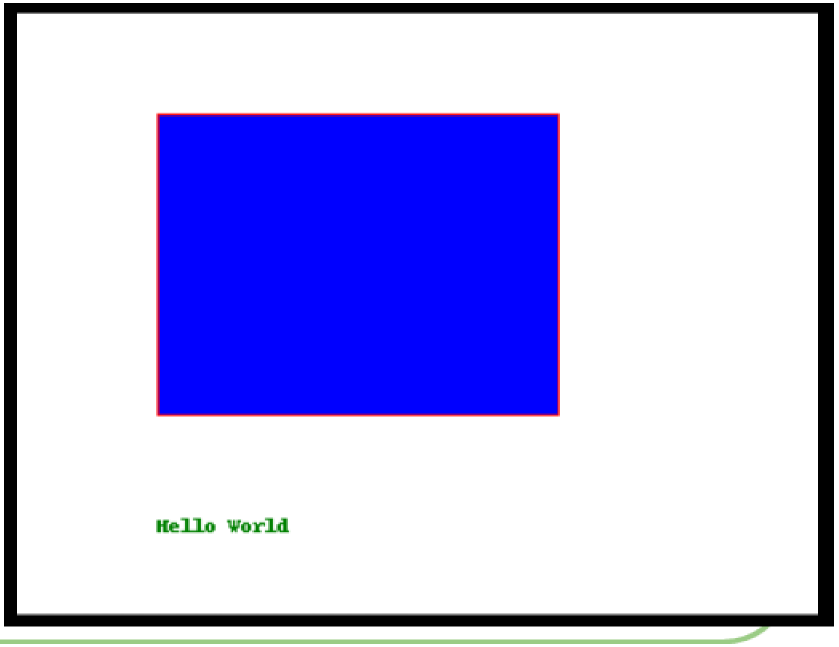


PIL.Image.FLIP_LEFT_RIGHT,
PIL.Image.FLIP_TOP_BOTTOM,
PIL.Image.ROTATE_90,
PIL.Image.ROTATE_180,
PIL.Image.ROTATE_270 or
PIL.Image.TRANSPOSE.



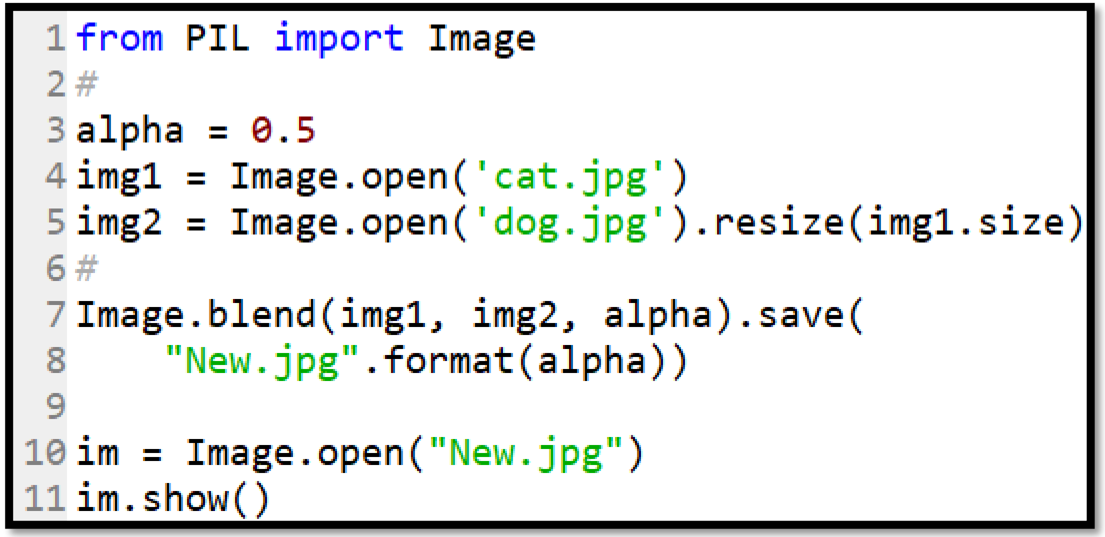
```python
from PIL import Image, ImageDraw

blank_image = Image.new('RGBA', (400, 300), 'white')
img_draw = ImageDraw.Draw(blank_image)
img_draw.rectangle((70, 50, 270, 200), outline='red', fill='blue')
img_draw.text((70, 250), 'Hello World', fill='green')
blank_image.save('drawn_image.bmp')

blank_image.show()
```

Output



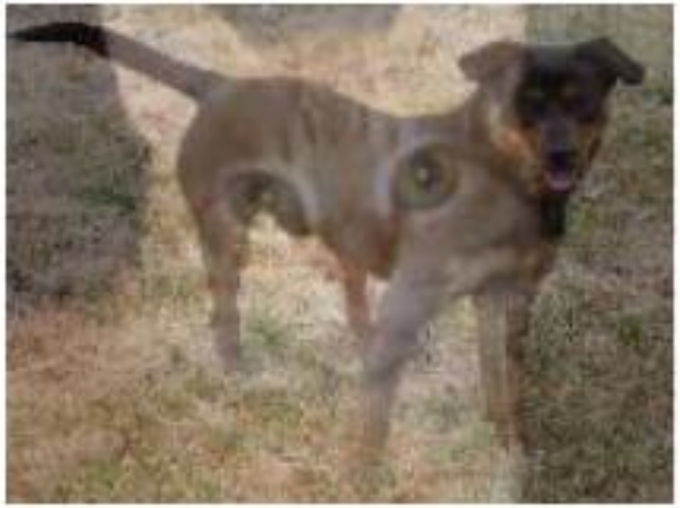
https://auth0.com/blog/image-processing-in-python-with-pillow/

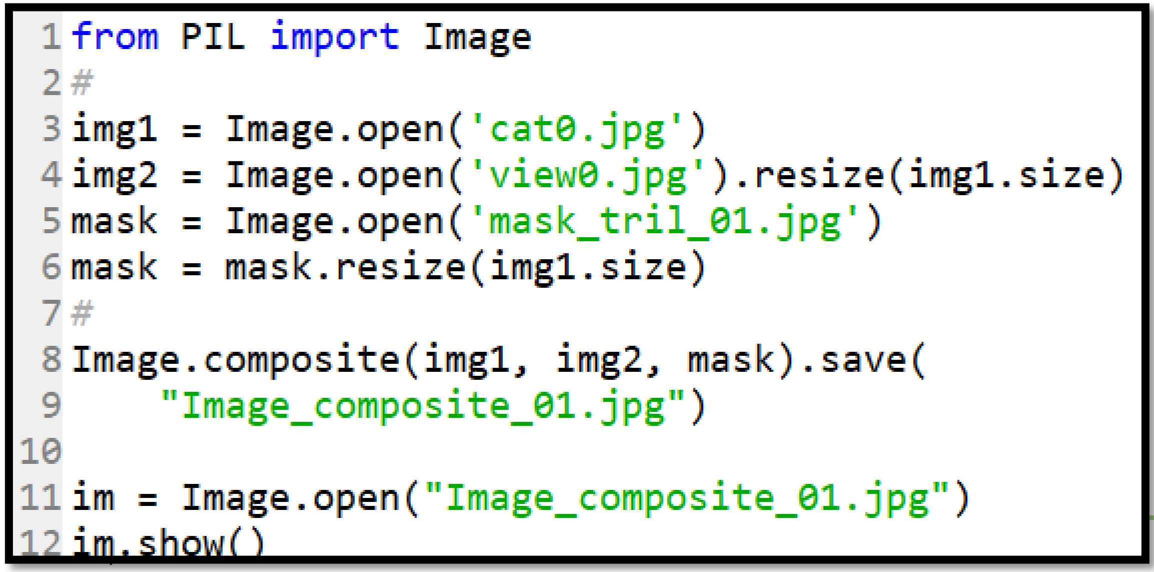
More functions



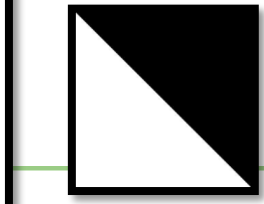
```python
from PIL import Image
#
alpha = 0.5
img1 = Image.open('cat.jpg')
img2 = Image.open('dog.jpg').resize(img1.size)
#
Image.blend(img1, img2, alpha).save(
    "New.jpg".format(alpha))

im = Image.open("New.jpg")
im.show()
```

Output

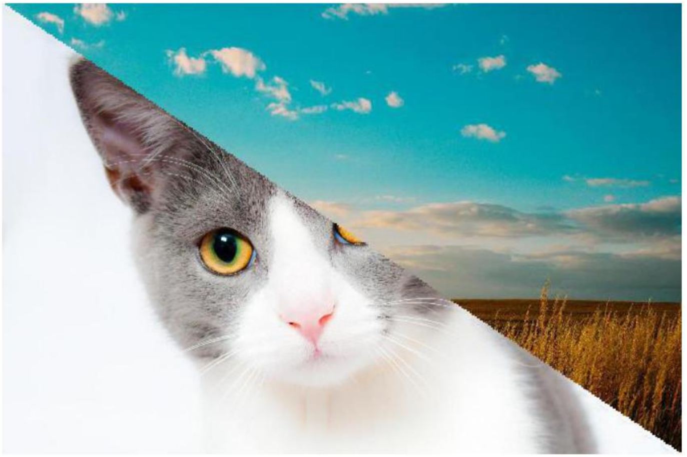




```python
from PIL import Image
#
img1 = Image.open('cat0.jpg')
img2 = Image.open('view0.jpg').resize(img1.size)
mask = Image.open('mask_tril_01.jpg')
mask = mask.resize(img1.size)
#
Image.composite(img1, img2, mask).save(
    "Image_composite_01.jpg")

im = Image.open("Image_composite_01.jpg")
im.show()
```

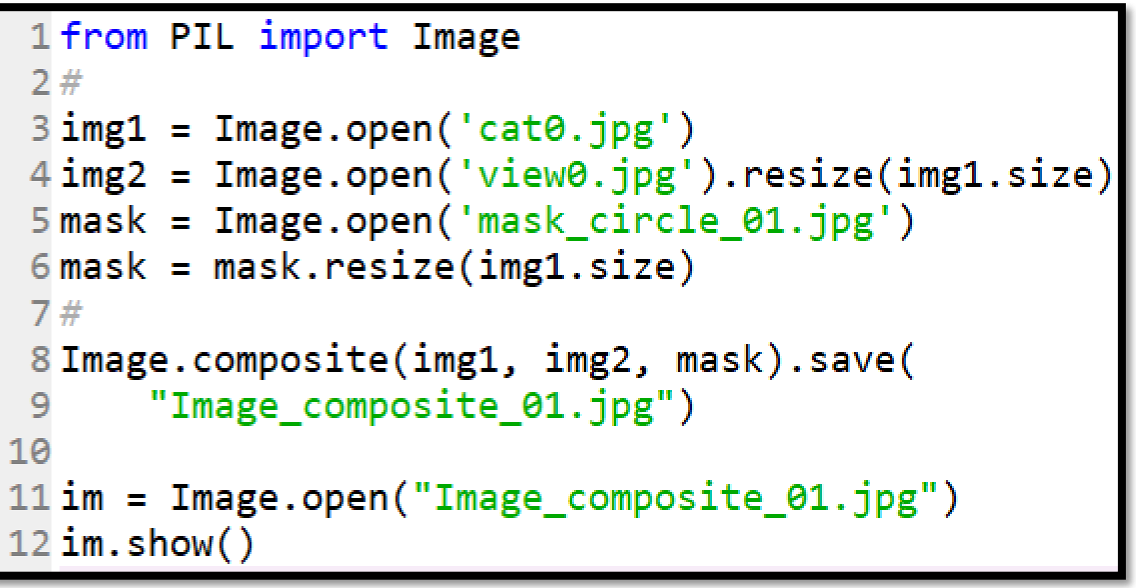


Output

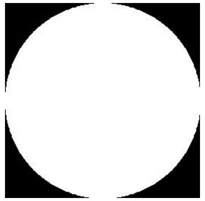


https://hhsprings.bitbucket.io/docs/programming/examples/python/PIL/Image__Functions.html#alpha-composite

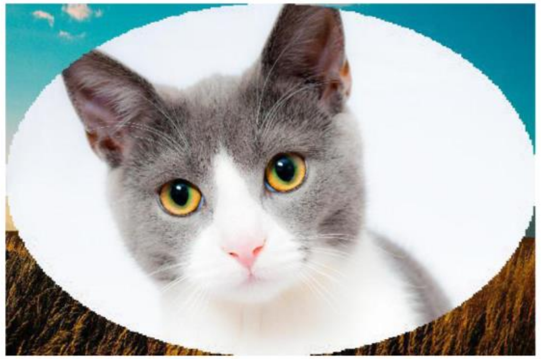
More functions

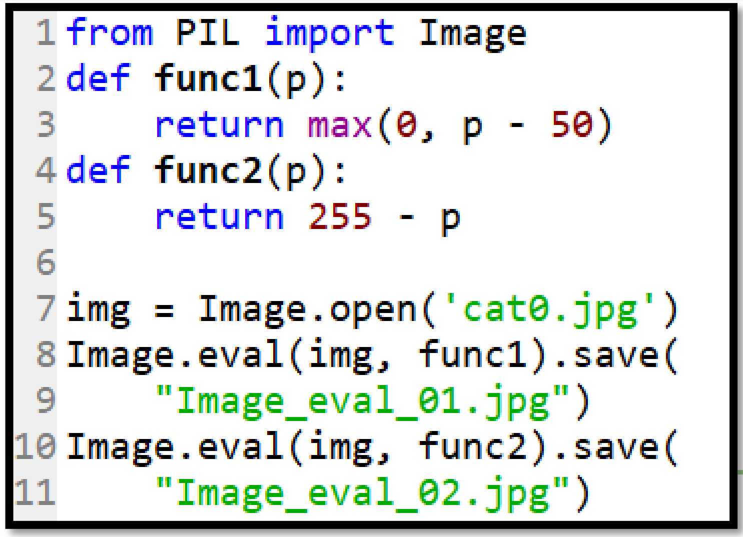


```python
from PIL import Image
#
img1 = Image.open('cat0.jpg')
img2 = Image.open('view0.jpg').resize(img1.size)
mask = Image.open('mask_circle_01.jpg')
mask = mask.resize(img1.size)
#
Image.composite(img1, img2, mask).save(
    "Image_composite_01.jpg")

im = Image.open("Image_composite_01.jpg")
im.show()
```



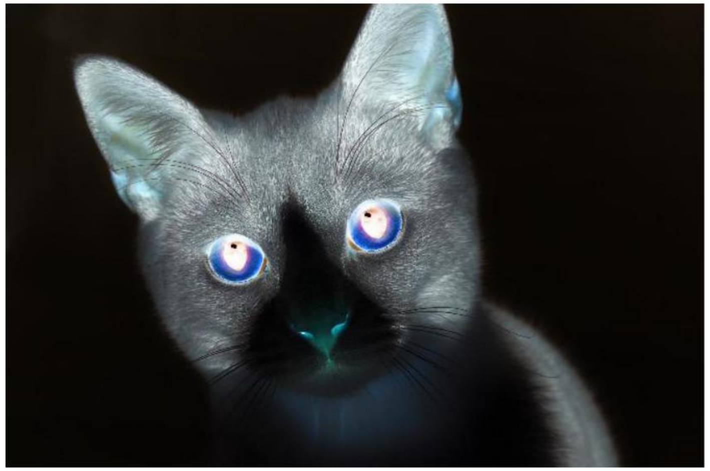
Output





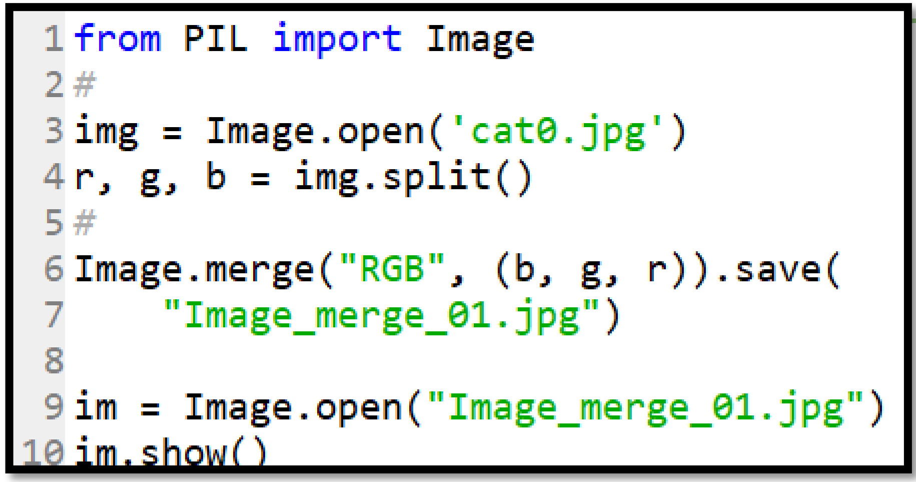
```python
from PIL import Image
def func1(p):
    return max(0, p - 50)
def func2(p):
    return 255 - p

img = Image.open('cat0.jpg')
Image.eval(img, func1).save(
    "Image_eval_01.jpg")
Image.eval(img, func2).save(
    "Image_eval_02.jpg")
```

Output



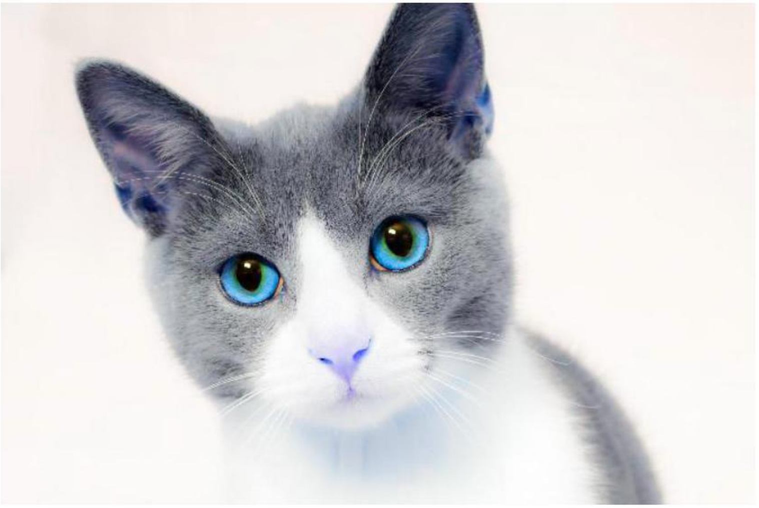
https://hhsprings.bitbucket.io/docs/programming/examples/python/PIL/Image__Functions.html#alpha-composite

More functions

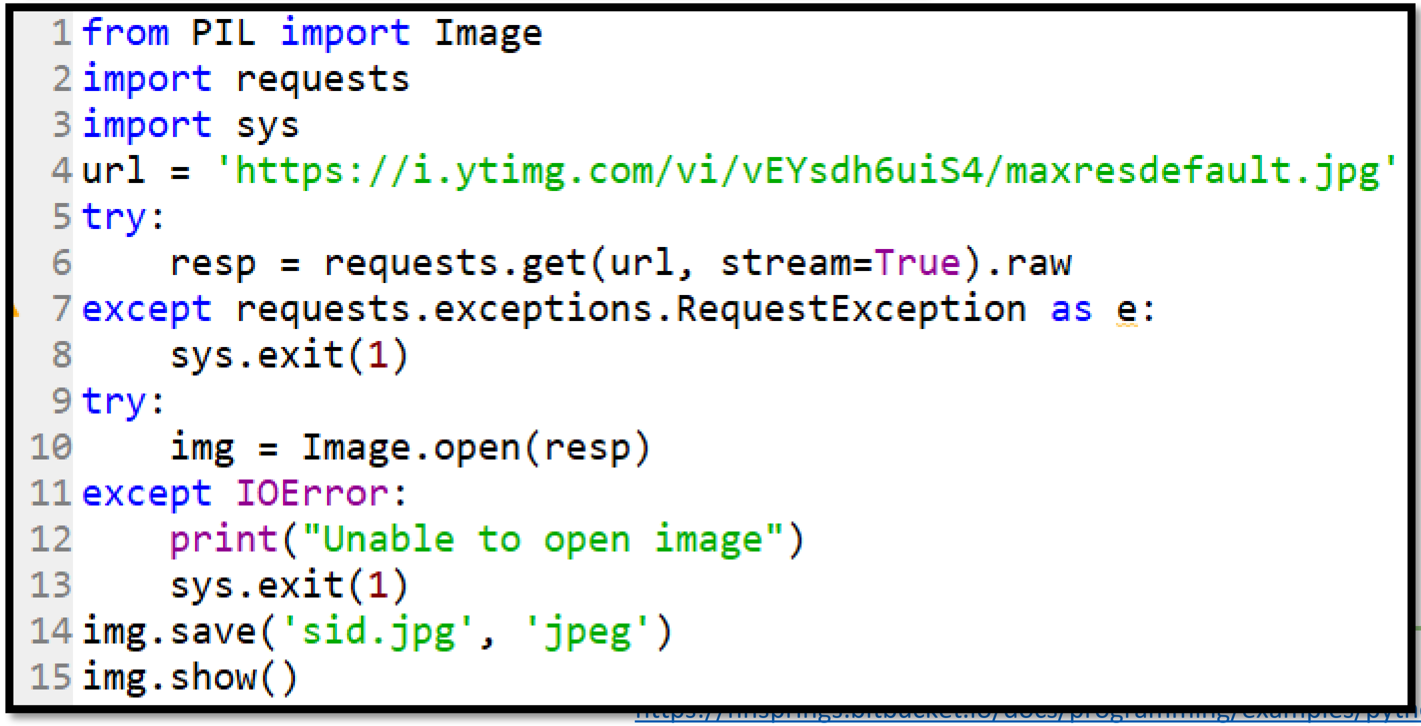


```python
from PIL import Image
#
img = Image.open('cat0.jpg')
r, g, b = img.split()
#
Image.merge("RGB", (b, g, r)).save(
    "Image_merge_01.jpg")

im = Image.open("Image_merge_01.jpg")
im.show()
```

(mode, bands)



Merge a set of single band images into a new multiband image.



```python
from PIL import Image
import requests
import sys
url = 'https://i.ytimg.com/vi/vEYsdh6uiS4/maxresdefault.jpg'
try:
    resp = requests.get(url, stream=True).raw
except requests.exceptions.RequestException as e:
    sys.exit(1)
try:
    img = Image.open(resp)
except IOError:
    print("Unable to open image")
    sys.exit(1)
img.save('sid.jpg', 'jpeg')
img.show()
```

https://hhsprings.bitbucket.io/docs/programming/examples/python/PIL/Image__Functions.html#alpha-composite

End of the lesson

Thank You

Master in Software Engineering

Hussam Hourani has over 25 years of Organizations Transformation, VROs, PMO, Large Scale and Enterprise Programs Global Delivery, Leadership, Business Development and Management Consulting. His client experience is wide ranging across many sectors but focuses on Performance Enhancement, Transformation, Enterprise Program Management, Artificial Intelligence and Data Science.