

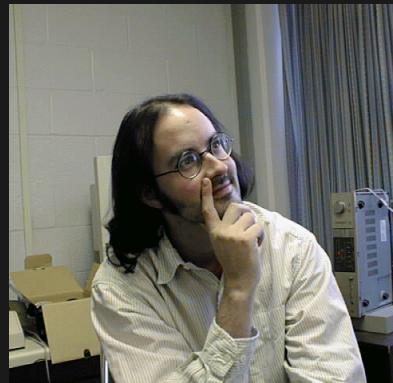


pillow

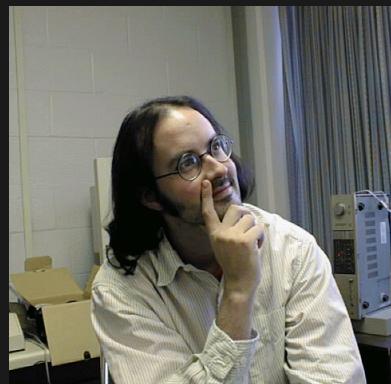
DC>>>
Python

Presented to DC Python Meetup & Austin Python
Meetup 4/16/24

ABOUT ME



ABOUT ME



Me circa 1998 😐

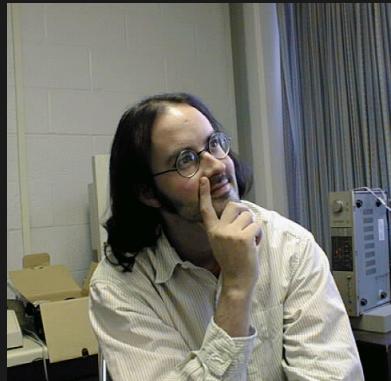
ABOUT ME



Me circa 1998 🤔

- Jeffrey A. Clark

ABOUT ME



Me circa 1998 🤔

- Jeffrey A. Clark
- Alex

ABOUT ME



Me circa 1998 🤔

- Jeffrey A. Clark
- Alex
- Bachelor of Science Computer Science 1998
Loyola University Maryland

ABOUT ME



ABOUT ME



Me circa 2008 😐

ABOUT ME



Me circa 2008 😐

- Python Pillow Creator 2010

ABOUT ME



Me circa 2008 😐

- Python Pillow Creator 2010
- PIL fork author

ABOUT ME



Me circa 2008 😐

- Python Pillow Creator 2010
- PIL fork author (↑same thing)

ABOUT ME



Me circa 2008 🤔

- Python Pillow Creator 2010
- PIL fork author (↑same thing)
- Tidelift Lifter 2019

ABOUT ME



Me circa 2008 😐

- Python Pillow Creator 2010
- PIL fork author (↑same thing)
- Tidelift Lifter 2019 (\$\$\$ for Pillow)

ABOUT ME



ABOUT ME



Me circa 2020

ABOUT ME



Me circa 2020

- Lifter

ABOUT ME



Me circa 2020

- Lifter 2019

ABOUT ME



Me circa 2020

- Lifter 2019 (actually late 2018)

ABOUT ME



Me circa 2020

- Lifter 2019 (actually late 2018)
- Lifter Advocate

ABOUT ME



Me circa 2020

- Lifter 2019 (actually late 2018)
- Lifter Advocate 2023

ABOUT ME



Me circa 2020

- Lifter 2019 (actually late 2018)
- Lifter Advocate 2023, 2024

ABOUT ME



Me circa 2020

- Lifter 2019 (actually late 2018)
- Lifter Advocate 2023, 2024
-

ABOUT ME



Me circa 2020

- Lifter 2019 (actually late 2018)
- Lifter Advocate 2023, 2024
- #opentowork

ABOUT ME



Me circa 2020

- Lifter 2019 (actually late 2018)
- Lifter Advocate 2023, 2024
- #opentowork (You should probably hire me.)

TIDELIFT

TIDELIFT

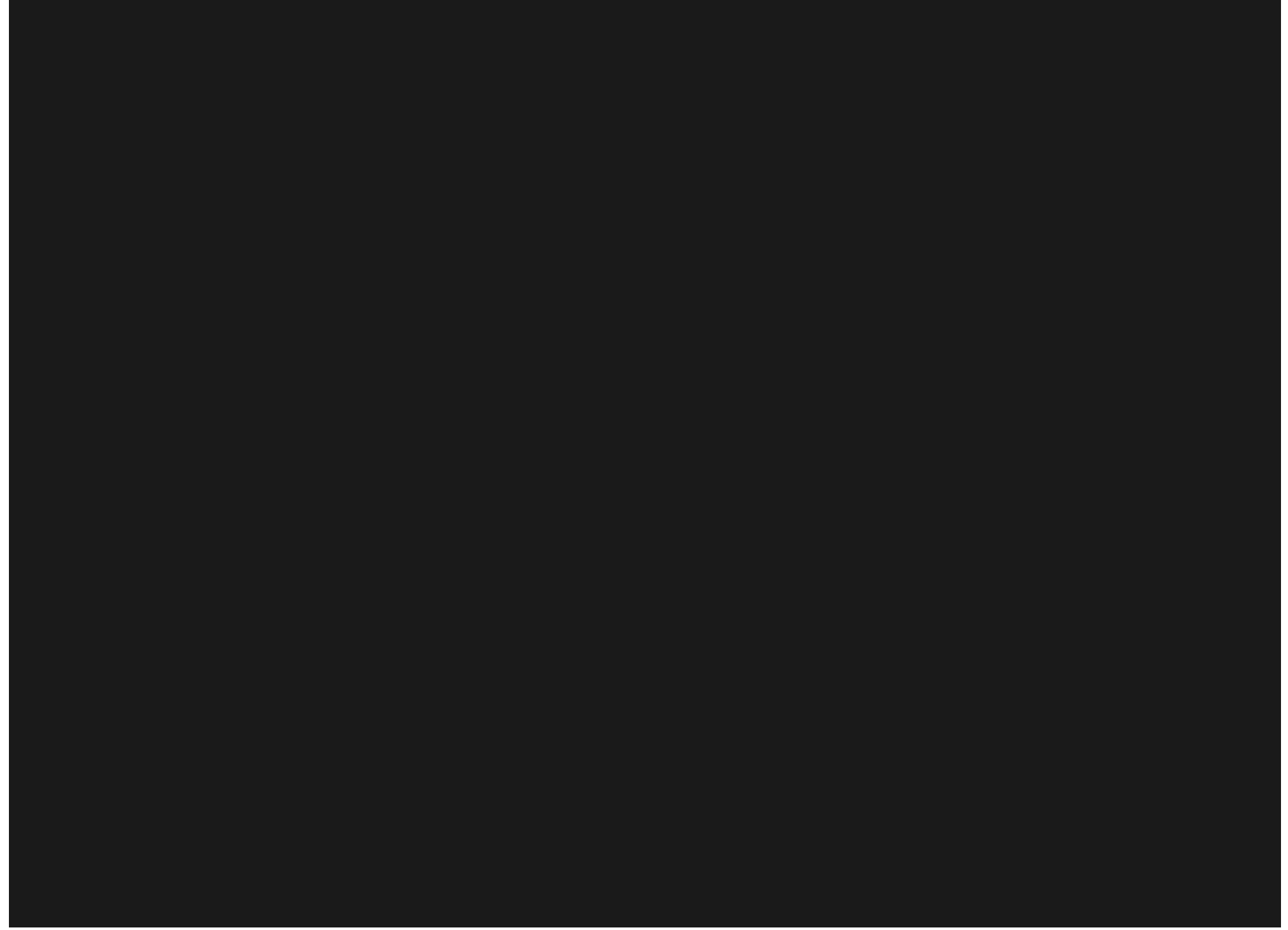


TIDELIFT

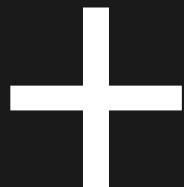


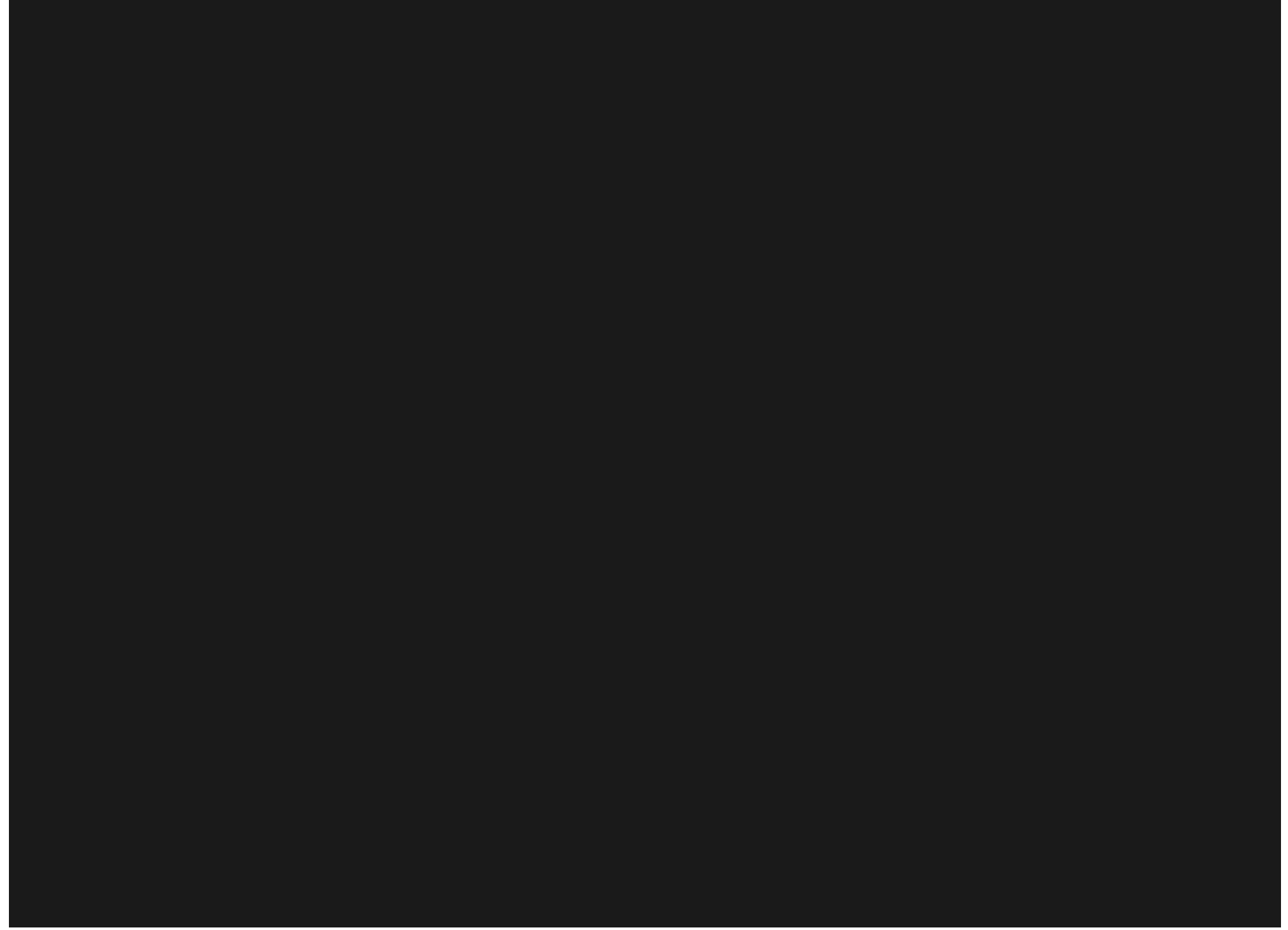
Let's pay the maintainers ✨

TIDELIFT

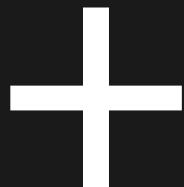


TIDELIFT





TIDELIFT



XZ

Let's talk about xz

Let's talk about xz

- <https://blog.tidelift.com/xz-tidelift-and-paying-the-maintainers>

Let's talk about xz

- <https://blog.tidelift.com/xz-tidelift-and-paying-the-maintainers>
- <https://blog.tidelift.com/the-roi-of-paying-open-source-maintainers-in-light-of-xz-utils-backdoor>

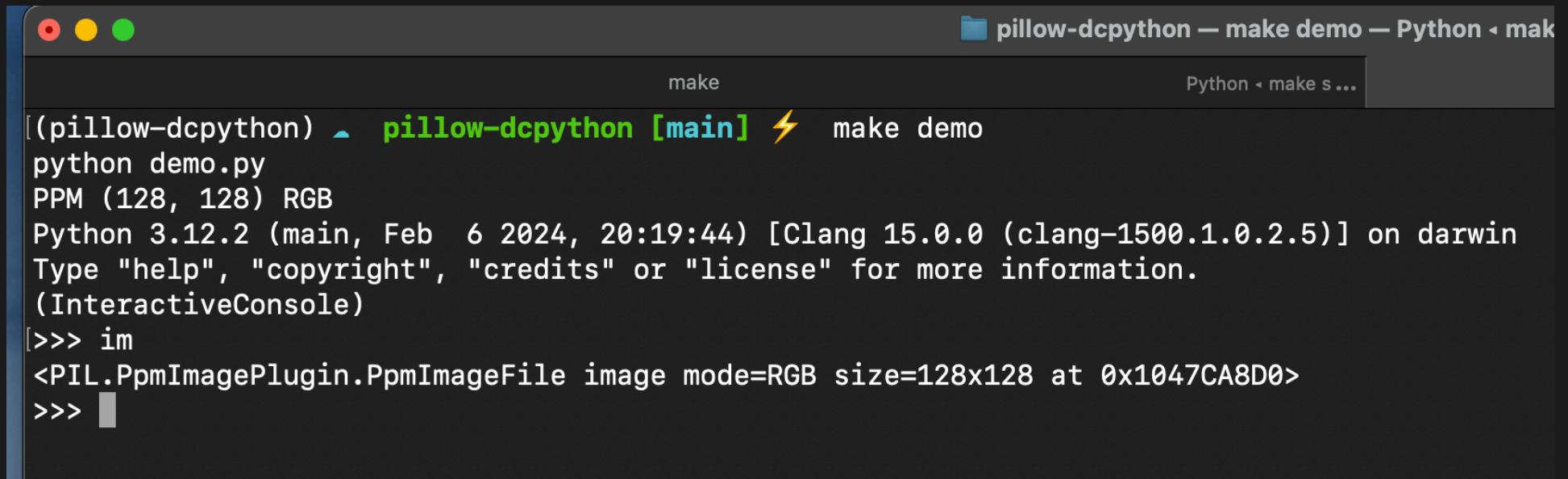
Let's talk about xz

- <https://blog.tidelift.com/xz-tidelift-and-paying-the-maintainers>
- <https://blog.tidelift.com/the-roi-of-paying-open-source-maintainers-in-light-of-xz-utils-backdoor>
- <https://blog.tidelift.com/maintainer-panel-hear-from-maintainers-in-a-post-xz-utils-backdoor-world>

Let's talk about Pillow

Let's talk about Pillow (Alex's Version)

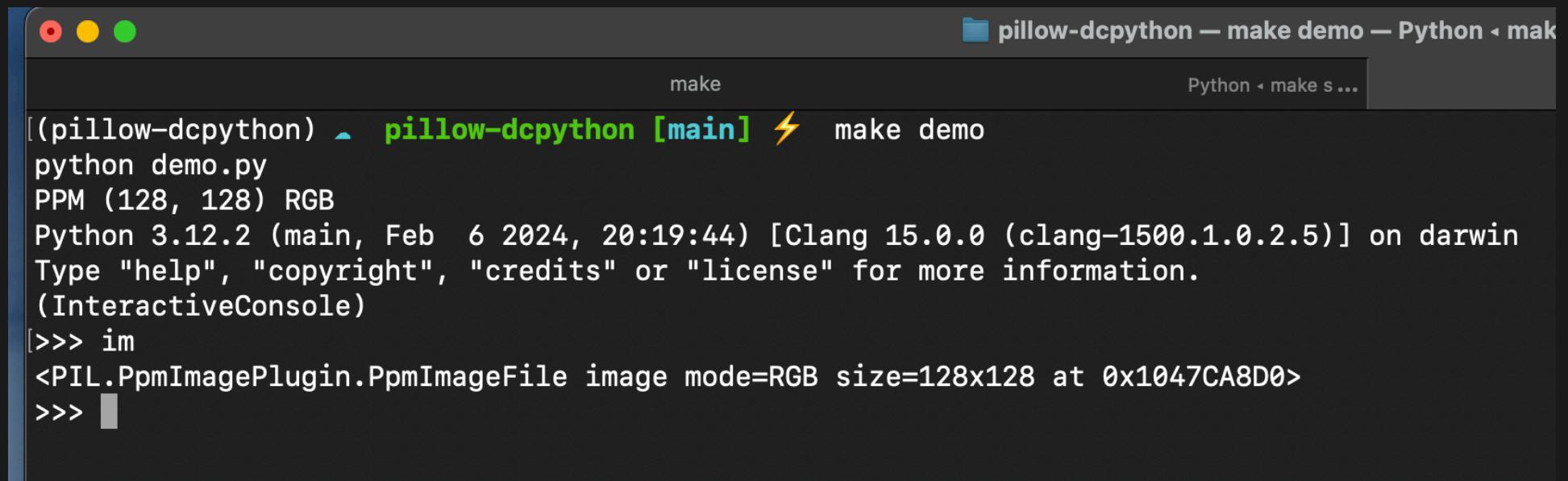
Let's talk about Pillow (Alex's Version)



A screenshot of a terminal window titled "pillow-dcpython — make demo — Python". The window shows the following Python session:

```
(pillow-dcpython) ~ pillow-dcpython [main] ⚡ make demo
python demo.py
PPM (128, 128) RGB
Python 3.12.2 (main, Feb 6 2024, 20:19:44) [Clang 15.0.0 (clang-1500.1.0.2.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
[>>> im
<PIL.PpmImagePlugin.PpmImageFile image mode=RGB size=128x128 at 0x1047CA8D0>
>>> ]
```

Let's talk about Pillow (Alex's Version)



A screenshot of a terminal window titled "pillow-dcpython — make demo — Python". The window shows the following Python session:

```
(pillow-dcpython) ~ pillow-dcpython [main] ⚡ make demo
python demo.py
PPM (128, 128) RGB
Python 3.12.2 (main, Feb 6 2024, 20:19:44) [Clang 15.0.0 (clang-1500.1.0.2.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
[>>> im
<PIL.PpmImagePlugin.PpmImageFile image mode=RGB size=128x128 at 0x1047CA8D0>
>>> ]
```

<https://github.com/python-pillow/pillow-demo>

Other Examples

Other Examples

- <https://github.com/janbodnar/pillow-examples>

Other Examples

- <https://github.com/janbodnar/pillow-examples>
- <https://realpython.com/image-processing-with-the-python-pillow-library/>

Other Examples

- <https://github.com/janbodnar/pillow-examples>
- <https://realpython.com/image-processing-with-the-python-pillow-library/>
- <https://hhsprings.bitbucket.io/docs/programming/examples/python/PIL/>

Other Examples

- <https://github.com/janbodnar/pillow-examples>
- <https://realpython.com/image-processing-with-the-python-pillow-library/>
- <https://hhsprings.bitbucket.io/docs/programming/examples/python/PIL/>
- <https://pillow.readthedocs.io/en/stable/handbook/index.html>

A screenshot of a GitHub repository page for 'pillow-examples' by janbodnar. The repository has 1 branch and 0 tags. It contains 15 commits from ade4014, mostly from 6 years ago. The commits include updates to README.md and various Python scripts like basic_info.py, blur_image.py, convert2png.py, crop_image.py, draw2image.py, grayscale.py, read_from_url.py, rotate_image.py, show_image.py, show_tkinter.py, and watermark.py. The repository has 2 forks and 2 stars. The 'About' section shows sources from the Pillow tutorial, a Readme file, activity, 2 stars, 2 watching, 4 forks, and a report repository link. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published. The 'Languages' section shows 100% Python.

Code

janbodnar / pillow-examples

Type to search

pillow-examples Public

Code

master 1 Branch 0 Tags

Go to file Add file

About

Sources from the Pillow tutorial

Readme Activity 2 stars 2 watching 4 forks Report repository

Releases

No releases published

Packages

No packages published

Languages

Python 100.0%

README

pillow-examples

Sources from the Pillow tutorial <http://zetcode.com/python/pillow/>

<https://zetcode.com/python/pillow/>

ZetCode

All Golang Python C# Java JavaScript Donate Subscribe

Where is that car? Is the battery dead?

Get it quick, know that it's ready to go.
Real-Time GPS Solution for your dealership

VANGUARD DEALER SERVICES
WAYPOINT GPS
Find Out More



Pillow

last modified January 29, 2024

Pillow tutorial shows how to use Pillow in Python to work with images. The sources are available at the author's Github [repository](#).

Advertisements



Pillow

Image To Sketch project using Python | Python ...

The screenshot shows a web browser displaying the Real Python website. The URL in the address bar is <https://realpython.com/image-processing-with-the-python-pillow/>. The page title is "Image Processing With the Python Pillow Library". The main content features a large illustration of a person standing next to a complex machine with various knobs, buttons, and labels like "PILLOW", "FILTERS", "SEGMENT", "SUPERPOSITION", and "KERNELS". Below the illustration, there's a "Table of Contents" section with a bullet point for "Basic Image Operations With the Python Pillow Library". To the right, there's a sidebar with a "FREE Email Series" section titled "Python Tricks" featuring a snippet of Python code for merging dictionaries. There are also buttons for "Email...", "Get Python Tricks »", and "No spam. Unsubscribe any time." At the bottom right, there's a "Browse Topics" section with categories like "Guided Learning Paths", "Basics", "Intermediate", and "Advanced". A footer navigation bar includes links for "api", "best-practices", "career", "community", "databases", and "data-science".

The screenshot shows a dark-themed web browser window displaying the [Pillow \(PIL Fork\) documentation handbook](https://pillow.readthedocs.io/en/stable/handbook/index.html). The URL is visible in the address bar. The page features a sidebar on the left with navigation links and a main content area on the right.

Pillow (PIL Fork)
10.3.0
documentation

Search:

Installation

Handbook

- Overview
- Tutorial
- Concepts
- Appendices

Reference

- Porting
- About
- Release Notes
- Deprecations and removals

ON-DEMAND PANEL

Predictions: What's in the crystal ball for open source software security in 2024?

WATCH NOW

RedMonk TIDELIFT

Predictions: Discover what's in the crystal ball for open source software security in 2024!

Handbook

- Overview
 - [Image Archives](#)
 - [Image Display](#)
 - [Image Processing](#)
- Tutorial
 - [Using the Image class](#)
 - [Reading and writing images](#)
 - [Cutting, pasting, and merging images](#)
 - [Geometrical transforms](#)
 - [Color transforms](#)
 - [Image enhancement](#)
 - [Image sequences](#)
 - [PostScript printing](#)
 - [More on reading images](#)
 - [Controlling the decoder](#)
- Concepts
 - [Bands](#)
 - [Modes](#)
 - [Size](#)
 - [Coordinate System](#)
 - [Palette](#)
 - [Info](#)
 - [Transparency](#)
 - [Orientation](#)
 - [Filters](#)
- Appendices
 - [Image file formats](#)
 - [Text anchors](#)
 - [Writing Your Own Image Plugin](#)
 - [Decoders](#)

Example #1 Open image and print info

Example #1 Open image and print info

```
>>> from PIL import Image
```

Example #1 Open image and print info

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")
```

Example #1 Open image and print info

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> print(im.format, im.size, im.mode)
```

Example #1 Open image and print info

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> print(im.format, im.size, im.mode)  
PPM (128, 128) RGB
```

Example #1 Open image and print info

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> print(im.format, im.size, im.mode)  
PPM (128, 128) RGB  
>>> im.show()
```

Example #1 Open image and print info

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> print(im.format, im.size, im.mode)  
PPM (128, 128) RGB  
>>> im.show()
```



Example #2 Rotate image 90 degrees clockwise (270 degrees counter clockwise)

Example #2 Rotate image 90 degrees clockwise (270 degrees counter clockwise)

```
>>> from PIL import Image
```

Example #2 Rotate image 90 degrees clockwise (270 degrees counter clockwise)

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")
```

Example #2 Rotate image 90 degrees clockwise (270 degrees counter clockwise)

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.rotate(270)
```

Example #2 Rotate image 90 degrees clockwise (270 degrees counter clockwise)

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.rotate(270)  
>>> im.save("rotated_hopper.jpg")
```

Example #2 Rotate image 90 degrees clockwise (270 degrees counter clockwise)

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.rotate(270)  
>>> im.save("rotated_hopper.jpg")  
>>> im.show()
```

Example #2 Rotate image 90 degrees clockwise (270 degrees counter clockwise)

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.rotate(270)  
>>> im.save("rotated_hopper.jpg")  
>>> im.show()
```



Example #3 Convert to jpeg

Example #3 Convert to jpeg

```
>>> from PIL import Image
```

Example #3 Convert to jpeg

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")
```

Example #3 Convert to jpeg

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im.save("hopper.jpg")
```

Example #3 Convert to jpeg

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im.save("hopper.jpg")  
$ file hopper.jpg
```

Example #3 Convert to jpeg

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im.save("hopper.jpg")  
$ file hopper.jpg  
hopper.jpg: JPEG image data, JFIF standard 1.01,  
aspect ratio, density 1x1, segment length 16, baseline,  
precision 8, 128x128, components 3
```

Example #4 Create thumbnails

Example #4 Create thumbnails

```
>>> from PIL import Image
```

Example #4 Create thumbnails

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")
```

Example #4 Create thumbnails

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im.thumbnail([64, 64])
```

Example #4 Create thumbnails

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im.thumbnail([64, 64])  
>>> im.save("thumbnail_hopper.jpg")
```

Example #4 Create thumbnails

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im.thumbnail([64, 64])  
>>> im.save("thumbnail_hopper.jpg")  
>>> im.show()
```

Example #4 Create thumbnails

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im.thumbnail([64, 64])  
>>> im.save("thumbnail_hopper.jpg")  
>>> im.show()
```



Example #5 Crop image

Example #5 Crop image

```
>>> from PIL import Image
```

Example #5 Crop image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")
```

Example #5 Crop image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.crop([0, 0, 64, 64])
```

Example #5 Crop image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.crop([0, 0, 64, 64])  
>>> im.save("cropped_hopper.jpg")
```

Example #5 Crop image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.crop([0, 0, 64, 64])  
>>> im.save("cropped_hopper.jpg")  
>>> im.show()
```

Example #5 Crop image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.crop([0, 0, 64, 64])  
>>> im.save("cropped_hopper.jpg")  
>>> im.show()
```



Example #6 Pasting image

Example #6 Pasting image

```
>>> from PIL import Image
```

Example #6 Pasting image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")
```

Example #6 Pasting image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> region = region.transpose(Image.Transpose.ROTATE_180)
```

Example #6 Pasting image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> region = region.transpose(Image.Transpose.ROTATE_180)  
>>> im.paste(region, (0, 0, 64, 64))
```

Example #6 Pasting image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> region = region.transpose(Image.Transpose.ROTATE_180)  
>>> im.paste(region, (0, 0, 64, 64))  
>>> im.save("pasted_hopper.jpg")
```

Example #6 Pasting image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> region = region.transpose(Image.Transpose.ROTATE_180)  
>>> im.paste(region, (0, 0, 64, 64))  
>>> im.save("pasted_hopper.jpg")  
>>> im.show()
```

Example #6 Pasting image

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> region = region.transpose(Image.Transpose.ROTATE_180)  
>>> im.paste(region, (0, 0, 64, 64))  
>>> im.save("pasted_hopper.jpg")  
>>> im.show()
```



Example #7 Roll image

Example #7 Roll image

```
>>> from PIL import Image
```

Example #7 Roll image

```
>>> from PIL import Image  
>>> from roll import roll
```

Example #7 Roll image

```
>>> from PIL import Image  
>>> from roll import roll  
>>> im = Image.open("hopper.ppm")
```

Example #7 Roll image

```
>>> from PIL import Image  
>>> from roll import roll  
>>> im = Image.open("hopper.ppm")  
>>> im = roll(im, 64)
```

Example #7 Roll image

```
>>> from PIL import Image  
>>> from roll import roll  
>>> im = Image.open("hopper.ppm")  
>>> im = roll(im, 64)  
>>> im.save("rolled_hopper.jpg")
```

Example #7 Roll image

```
>>> from PIL import Image  
>>> from roll import roll  
>>> im = Image.open("hopper.ppm")  
>>> im = roll(im, 64)  
>>> im.save("rolled_hopper.jpg")  
>>> im.show()
```

Example #7 Roll image

```
>>> from PIL import Image  
>>> from roll import roll  
>>> im = Image.open("hopper.ppm")  
>>> im = roll(im, 64)  
>>> im.save("rolled_hopper.jpg")  
>>> im.show()
```



Example #7 roll.py

Example #7 roll.py

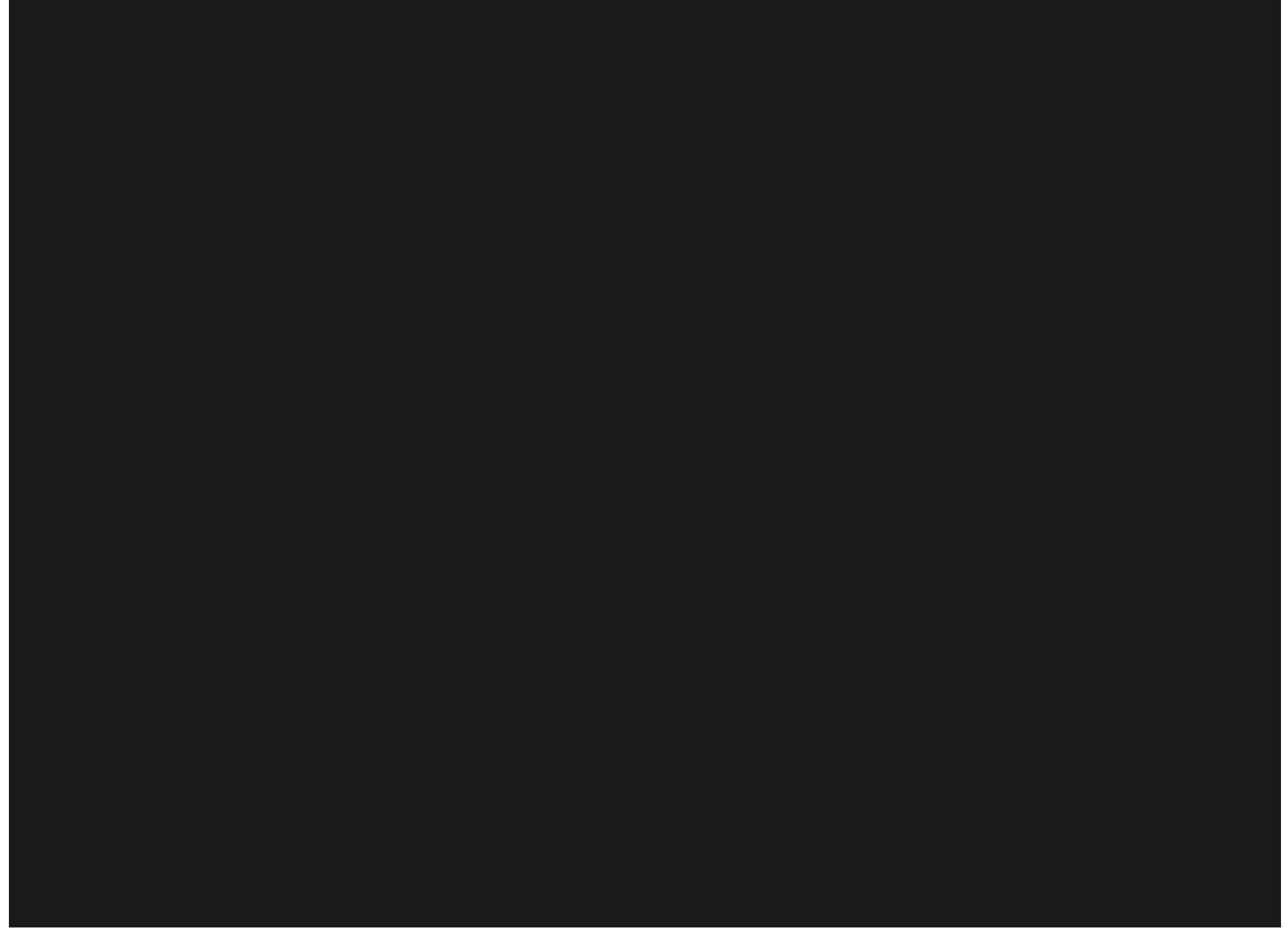
```
def roll(im, delta):
    """Roll an image sideways."""
    xsize, ysize = im.size
    delta = delta % xsize
    if delta == 0:
        return im
    part1 = im.crop((0, 0, delta, ysize))
    part2 = im.crop((delta, 0, xsize, ysize))
    im.paste(part1, (xsize - delta, 0, xsize, ysize))
    im.paste(part2, (0, 0, xsize - delta, ysize))
    return im
```

Example #8 Merge images



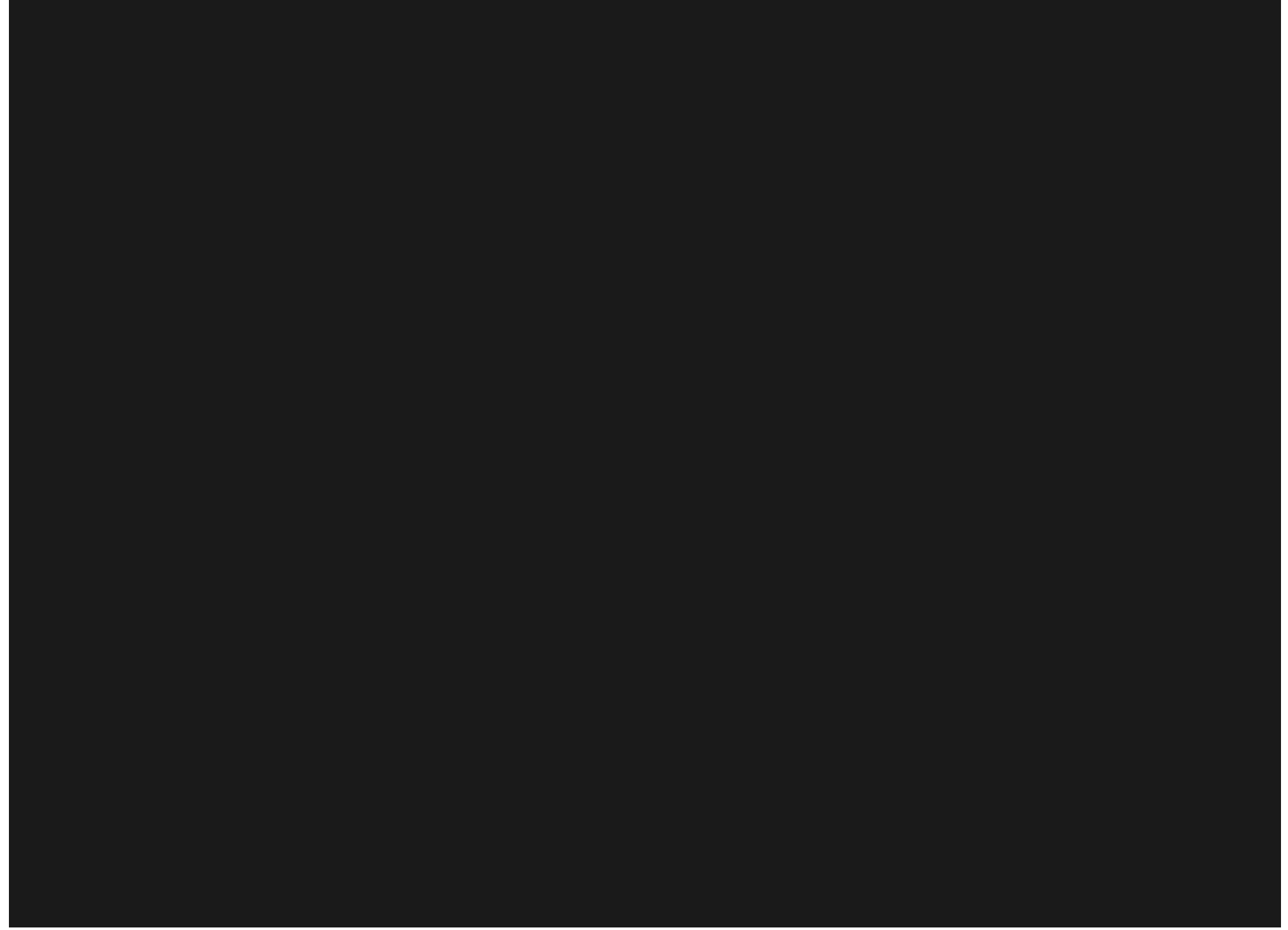
Example #8 Merge images

```
>>> from PIL import Image
```



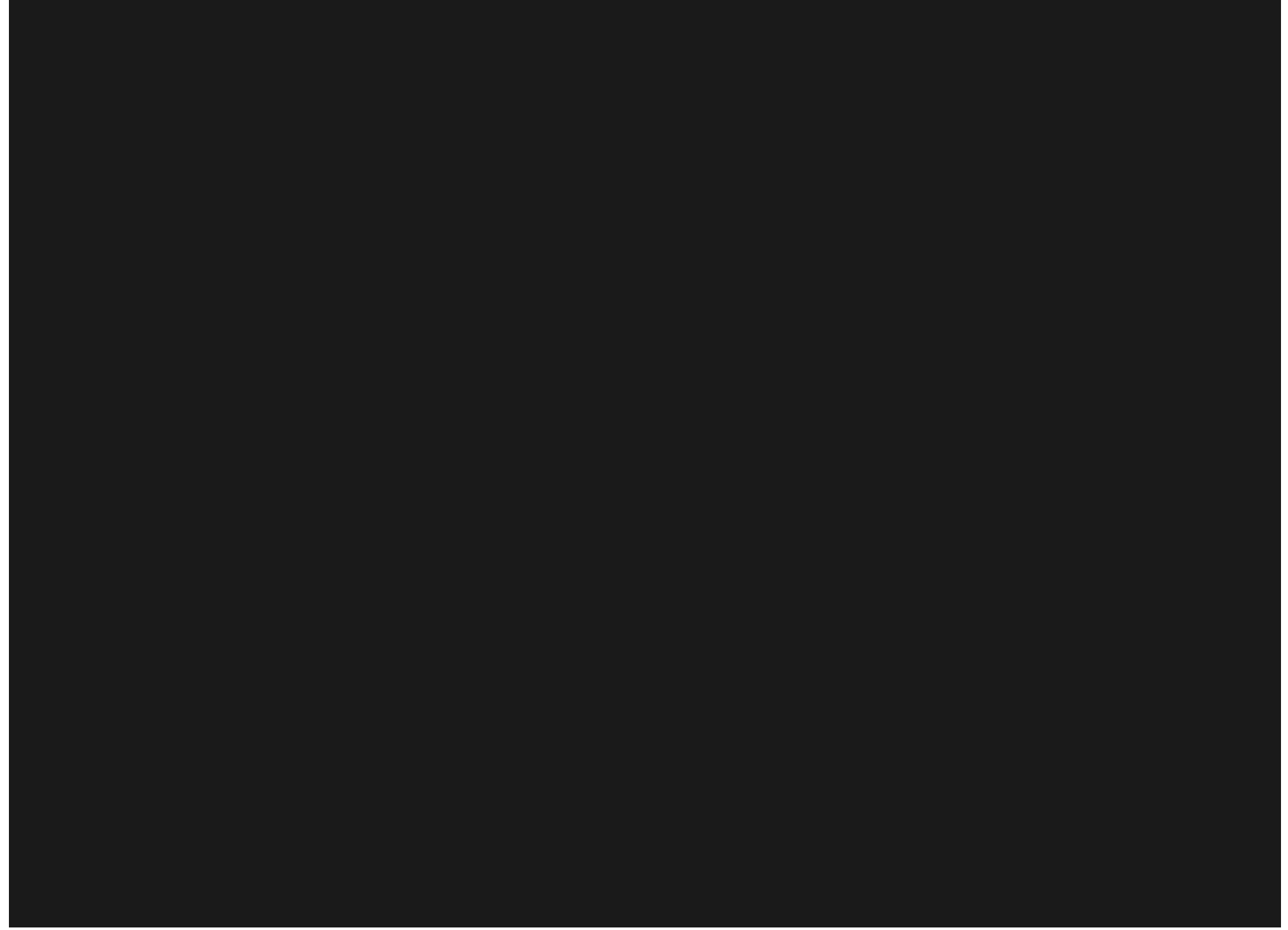
Example #8 Merge images

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")
```



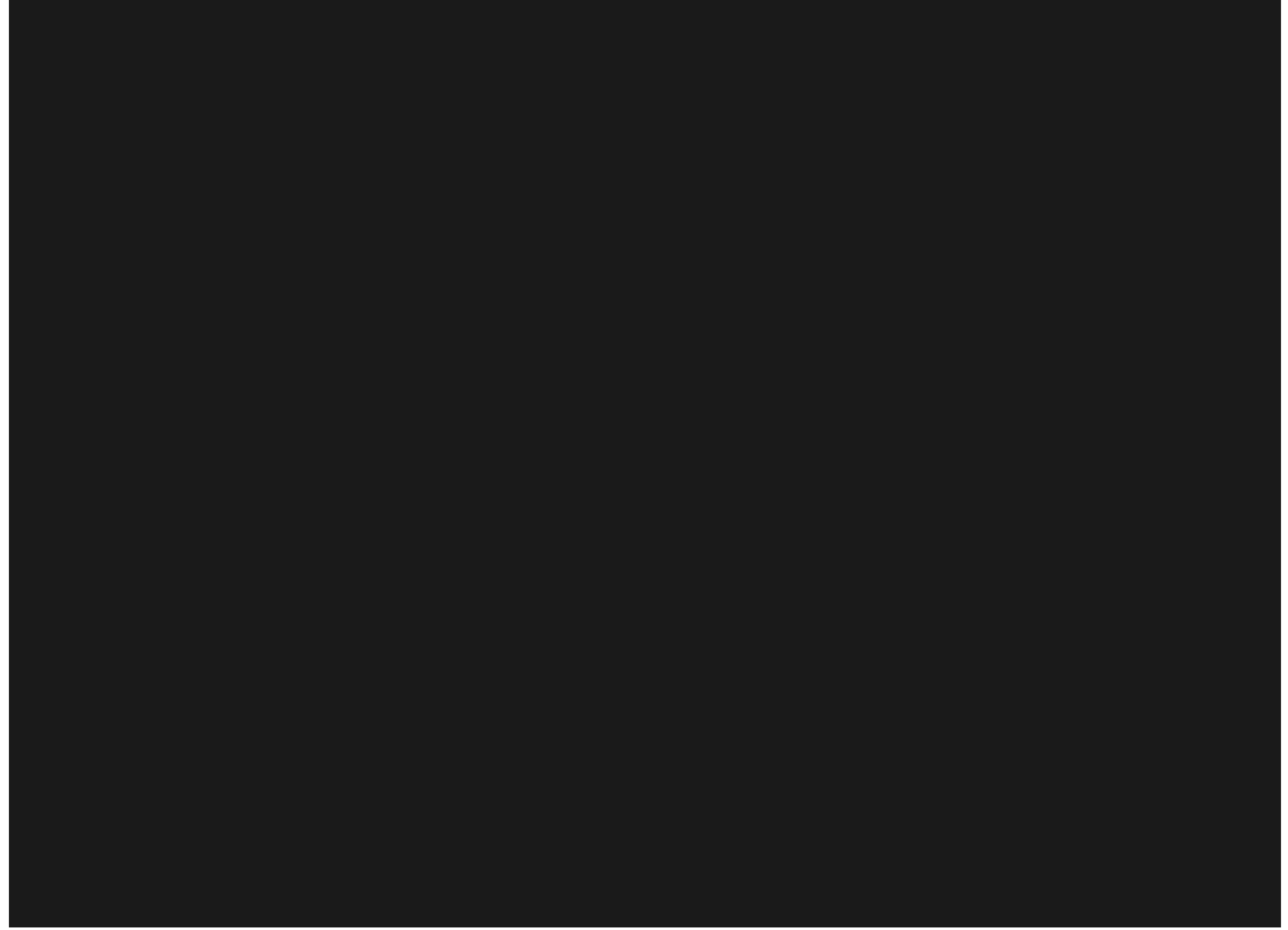
Example #8 Merge images

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")
```



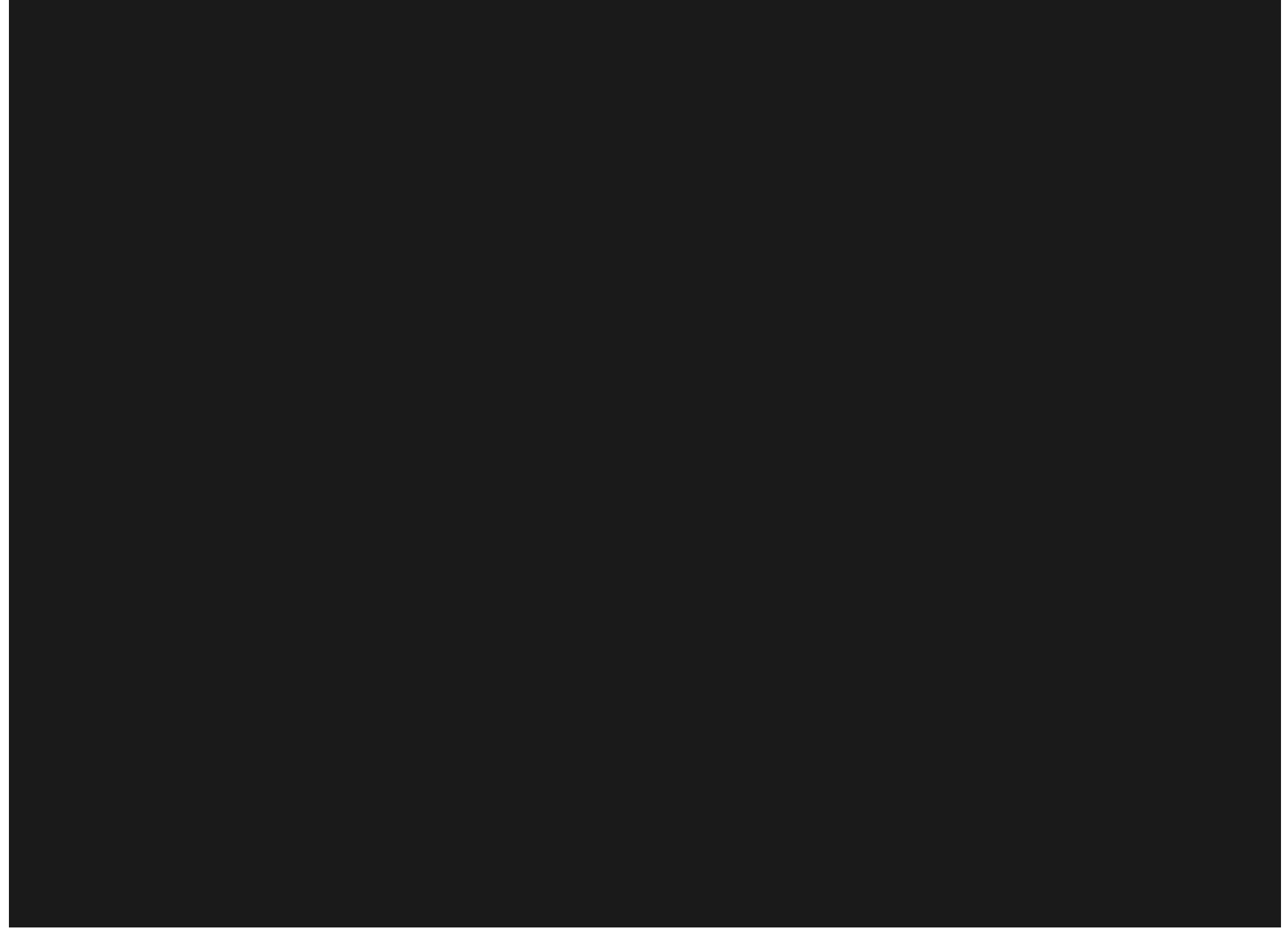
Example #8 Merge images

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)
```



Example #8 Merge images

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_hopper.png")
```



Example #8 Merge images

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_hopper.png")  
>>> im.show()
```



Example #8 Merge images

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_hopper.png")  
>>> im.show()
```



Example #8 Merge images (and resize)

Example #8 Merge images (and resize)

```
>>> from PIL import Image
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_resized_hopper.png")
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_resized_hopper.png")  
>>> im = Image.open("merged_resized_hopper.png")
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_resized_hopper.png")  
>>> im = Image.open("merged_resized_hopper.png")  
>>> im = im.resize([128, 128])
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_resized_hopper.png")  
>>> im = Image.open("merged_resized_hopper.png")  
>>> im = im.resize([128, 128])  
>>> im.save("merged_resized_hopper.png")
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_resized_hopper.png")  
>>> im = Image.open("merged_resized_hopper.png")  
>>> im = im.resize([128, 128])  
>>> im.save("merged_resized_hopper.png")  
>>> im.show()
```

Example #8 Merge images (and resize)

```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_resized_hopper.png")  
>>> im = Image.open("merged_resized_hopper.png")  
>>> im = im.resize([128, 128])  
>>> im.save("merged_resized_hopper.png")  
>>> im.show()
```



Example #9 Split and merge bands

Example #9 Split and merge bands

```
>>> from PIL import Image
```

Example #9 Split and merge bands

```
>>> from PIL import Image  
>>> r, g, b = im.split()
```

Example #9 Split and merge bands

```
>>> from PIL import Image  
>>> r, g, b = im.split()  
>>> im = Image.merge("RGB", (b, g, r))
```

Example #9 Split and merge bands

```
>>> from PIL import Image  
>>> r, g, b = im.split()  
>>> im = Image.merge("RGB", (b, g, r))  
>>> im.save("rebanded_hopper.jpg")
```

Example #9 Split and merge bands

```
>>> from PIL import Image  
>>> r, g, b = im.split()  
>>> im = Image.merge("RGB", (b, g, r))  
>>> im.save("rebanded_hopper.jpg")  
>>> im.show()
```

Example #9 Split and merge bands

```
>>> from PIL import Image  
>>> r, g, b = im.split()  
>>> im = Image.merge("RGB", (b, g, r))  
>>> im.save("rebanded_hopper.jpg")  
>>> im.show()
```



Example #10 Create JPEG thumbnail with resize

Example #10 Create JPEG thumbnail with resize

```
>>> from PIL import Image
```

Example #10 Create JPEG thumbnail with resize

```
>>> from PIL import Image  
>>> im = im.resize([64, 64])
```

Example #10 Create JPEG thumbnail with resize

```
>>> from PIL import Image  
>>> im = im.resize([64, 64])  
>>> im.save("resized_hopper.jpg")
```

Example #10 Create JPEG thumbnail with resize

```
>>> from PIL import Image  
>>> im = im.resize([64, 64])  
>>> im.save("resized_hopper.jpg")  
>>> im.show()
```

Example #10 Create JPEG thumbnail with resize

```
>>> from PIL import Image  
>>> im = im.resize([64, 64])  
>>> im.save("resized_hopper.jpg")  
>>> im.show()
```



Example #11 Transpose image left to right

Example #11 Transpose image left to right

```
>>> from PIL import Image
```

Example #11 Transpose image left to right

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_LEFT_RIGHT)
```

Example #11 Transpose image left to right

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_LEFT_RIGHT)  
>>> im.save("flip_left_right_hopper.jpg")
```

Example #11 Transpose image left to right

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_LEFT_RIGHT)  
>>> im.save("flip_left_right_hopper.jpg")  
>>> im.show()
```

Example #11 Transpose image left to right

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_LEFT_RIGHT)  
>>> im.save("flip_left_right_hopper.jpg")  
>>> im.show()
```



Example #12 Transpose image top to bottom

Example #12 Transpose image top to bottom

```
>>> from PIL import Image
```

Example #12 Transpose image top to bottom

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_TOP_BOTTOM)
```

Example #12 Transpose image top to bottom

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_TOP_BOTTOM)  
>>> im.save("flip_top_bottom_hopper.jpg")
```

Example #12 Transpose image top to bottom

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_TOP_BOTTOM)  
>>> im.save("flip_top_bottom_hopper.jpg")  
>>> im.show()
```

Example #12 Transpose image top to bottom

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.FLIP_TOP_BOTTOM)  
>>> im.save("flip_top_bottom_hopper.jpg")  
>>> im.show()
```



Example #13 Rotate image 90 degrees with transpose

Example #13 Rotate image 90 degrees with transpose

```
>>> from PIL import Image
```

Example #13 Rotate image 90 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_90)
```

Example #13 Rotate image 90 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_90)  
>>> im.save("rotated_hopper_90.jpg")
```

Example #13 Rotate image 90 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_90)  
>>> im.save("rotated_hopper_90.jpg")  
>>> im.show()
```

Example #13 Rotate image 90 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_90)  
>>> im.save("rotated_hopper_90.jpg")  
>>> im.show()
```



Example #14 Rotate image 180 degrees with transpose

Example #14 Rotate image 180 degrees with transpose

```
>>> from PIL import Image
```

Example #14 Rotate image 180 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_180)
```

Example #14 Rotate image 180 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_180)  
>>> im.save("rotated_hopper_180.jpg")
```

Example #14 Rotate image 180 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_180)  
>>> im.save("rotated_hopper_180.jpg")  
>>> im.show()
```

Example #14 Rotate image 180 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_180)  
>>> im.save("rotated_hopper_180.jpg")  
>>> im.show()
```



Example #15 Rotate image 270 degrees with transpose

Example #15 Rotate image 270 degrees with transpose

```
>>> from PIL import Image
```

Example #15 Rotate image 270 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_270)
```

Example #15 Rotate image 270 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_270)  
>>> im.save("rotated_hopper_270.jpg")
```

Example #15 Rotate image 270 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_270)  
>>> im.save("rotated_hopper_270.jpg")  
>>> im.show()
```

Example #15 Rotate image 270 degrees with transpose

```
>>> from PIL import Image  
>>> im = im.transpose(Image.Transpose.ROTATE_270)  
>>> im.save("rotated_hopper_270.jpg")  
>>> im.show()
```



Example #16 Relative resize image with contain

Example #16 Relative resize image with contain

```
>>> from PIL import ImageOps
```

Example #16 Relative resize image with contain

```
>>> from PIL import ImageOps  
>>> ImageOps.contain(im, (100, 150)).save(
```

Example #16 Relative resize image with contain

```
>>> from PIL import ImageOps  
>>> ImageOps.contain(im, (100, 150)).save(  
>>>     "contained_hopper.png")
```

Example #16 Relative resize image with contain

```
>>> from PIL import ImageOps  
>>> ImageOps.contain(im, (100, 150)).save(  
>>>     "contained_hopper.png")  
$ file contained_hopper.png
```

Example #16 Relative resize image with contain

```
>>> from PIL import ImageOps  
>>> ImageOps.contain(im, (100, 150)).save(  
>>>     "contained_hopper.png")  
$ file contained_hopper.png  
contained_hopper.png: PNG image data,  
100 x 100, 8-bit/color RGB, non-interlaced
```

Example #16 Relative resize image with contain

```
>>> from PIL import ImageOps  
>>> ImageOps.contain(im, (100, 150)).save(  
>>>     "contained_hopper.png")  
$ file contained_hopper.png  
contained_hopper.png: PNG image data,  
100 x 100, 8-bit/color RGB, non-interlaced
```



Example #17 Relative resize image with cover

Example #17 Relative resize image with cover

```
>>> from PIL import ImageOps
```

Example #17 Relative resize image with cover

```
>>> from PIL import ImageOps  
>>> ImageOps.cover(im, (100, 150)).save("covered_hopper.png")
```

Example #17 Relative resize image with cover

```
>>> from PIL import ImageOps  
>>> ImageOps.cover(im, (100, 150)).save("covered_hopper.png")  
$ file covered_hopper.png
```

Example #17 Relative resize image with cover

```
>>> from PIL import ImageOps  
>>> ImageOps.cover(im, (100, 150)).save("covered_hopper.png")  
$ file covered_hopper.png  
covered_hopper.png: PNG image data,  
150 x 150, 8-bit/color RGB, non-interlaced
```

Example #17 Relative resize image with cover

```
>>> from PIL import ImageOps  
>>> ImageOps.cover(im, (100, 150)).save("covered_hopper.png")  
$ file covered_hopper.png  
covered_hopper.png: PNG image data,  
150 x 150, 8-bit/color RGB, non-interlaced
```



Example #18 Relative resize image with fit

Example #18 Relative resize image with fit

```
>>> from PIL import ImageOps
```

Example #18 Relative resize image with fit

```
>>> from PIL import ImageOps  
>>> ImageOps.fit(im, (100, 150)).save("fitted_hopper.png")
```

Example #18 Relative resize image with fit

```
>>> from PIL import ImageOps  
>>> ImageOps.fit(im, (100, 150)).save("fitted_hopper.png")  
$ file fitted_hopper.png
```

Example #18 Relative resize image with fit

```
>>> from PIL import ImageOps  
>>> ImageOps.fit(im, (100, 150)).save("fitted_hopper.png")  
$ file fitted_hopper.png  
fitted_hopper.png: PNG image data,  
100 x 150, 8-bit/color RGB, non-interlaced
```

Example #18 Relative resize image with fit

```
>>> from PIL import ImageOps  
>>> ImageOps.fit(im, (100, 150)).save("fitted_hopper.png")  
$ file fitted_hopper.png  
fitted_hopper.png: PNG image data,  
100 x 150, 8-bit/color RGB, non-interlaced
```



Example #19 Relative resize image with pad

Example #19 Relative resize image with pad

```
>>> from PIL import ImageOps
```

Example #19 Relative resize image with pad

```
>>> from PIL import ImageOps  
>>> ImageOps.pad(im, (100, 150), color="#f00").save("padded_ho
```

Example #19 Relative resize image with pad

```
>>> from PIL import ImageOps  
>>> ImageOps.pad(im, (100, 150), color="#f00").save("padded_ho  
$ file padded_hopper.png
```

Example #19 Relative resize image with pad

```
>>> from PIL import ImageOps  
>>> ImageOps.pad(im, (100, 150), color="#f00").save("padded_ho  
$ file padded_hopper.png  
padded_hopper.png: PNG image data,  
100 x 150, 8-bit/color RGB, non-interlaced
```

Example #19 Relative resize image with pad

```
>>> from PIL import ImageOps  
>>> ImageOps.pad(im, (100, 150), color="#f00").save("padded_ho  
$ file padded_hopper.png  
padded_hopper.png: PNG image data,  
100 x 150, 8-bit/color RGB, non-interlaced
```



Example #20 Convert mode

Example #20 Convert mode

```
>>> from PIL import Image
```

Example #20 Convert mode

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")
```

Example #20 Convert mode

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.convert("L")
```

Example #20 Convert mode

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.convert("L")  
>>> im.save("converted_hopper.jpg")
```

Example #20 Convert mode

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.convert("L")  
>>> im.save("converted_hopper.jpg")  
>>> im.show()
```

Example #20 Convert mode

```
>>> from PIL import Image  
>>> im = Image.open("hopper.ppm")  
>>> im = im.convert("L")  
>>> im.save("converted_hopper.jpg")  
>>> im.show()
```



Example #21 Image enhancement

Example #21 Image enhancement

```
>>> from PIL import ImageFilter
```

Example #21 Image enhancement

```
>>> from PIL import ImageFilter  
>>> im = im.filter(ImageFilter.DETAIL)
```

Example #21 Image enhancement

```
>>> from PIL import ImageFilter  
>>> im = im.filter(ImageFilter.DETAIL)  
>>> im.save("enhanced_hopper.jpg")
```

Example #21 Image enhancement

```
>>> from PIL import ImageFilter  
>>> im = im.filter(ImageFilter.DETAIL)  
>>> im.save("enhanced_hopper.jpg")  
>>> im.show()
```

Example #21 Image enhancement

```
>>> from PIL import ImageFilter  
>>> im = im.filter(ImageFilter.DETAIL)  
>>> im.save("enhanced_hopper.jpg")  
>>> im.show()
```



Example #22 Point operations

Example #22 Point operations

```
>>> from PIL import Image
```

Example #22 Point operations

```
>>> from PIL import Image  
>>> im = im.point(lambda i: i * 20)
```

Example #22 Point operations

```
>>> from PIL import Image  
>>> im = im.point(lambda i: i * 20)  
>>> im.save("transformed_hopper.jpg")
```

Example #22 Point operations

```
>>> from PIL import Image  
>>> im = im.point(lambda i: i * 20)  
>>> im.save("transformed_hopper.jpg")  
>>> im.show()
```

Example #22 Point operations

```
>>> from PIL import Image  
>>> im = im.point(lambda i: i * 20)  
>>> im.save("transformed_hopper.jpg")  
>>> im.show()
```



Example #23 Process bands

Example #23 Process bands

```
>>> from PIL import Image
```

Example #23 Process bands

```
>>> from PIL import Image  
>>> source = im.split()
```

Example #23 Process bands

```
>>> from PIL import Image  
>>> source = im.split()  
>>> r, g, b = 0, 1, 2
```

Example #23 Process bands

```
>>> from PIL import Image  
>>> source = im.split()  
>>> r, g, b = 0, 1, 2  
>>> mask = source[r].point(lambda i: i < 100 and 255)
```

Example #23 Process bands

```
>>> from PIL import Image  
>>> source = im.split()  
>>> r, g, b = 0, 1, 2  
>>> mask = source[r].point(lambda i: i < 100 and 255)  
>>> out = source[g].point(lambda i: i * 0.7)
```

Example #23 Process bands

```
>>> from PIL import Image  
>>> source = im.split()  
>>> r, g, b = 0, 1, 2  
>>> mask = source[r].point(lambda i: i < 100 and 255)  
>>> out = source[g].point(lambda i: i * 0.7)  
>>> source[g].paste(out, None, mask)
```

Example #23 Process bands

```
>>> from PIL import Image  
>>> source = im.split()  
>>> r, g, b = 0, 1, 2  
>>> mask = source[r].point(lambda i: i < 100 and 255)  
>>> out = source[g].point(lambda i: i * 0.7)  
>>> source[g].paste(out, None, mask)  
>>> im = Image.merge(im.mode, source)
```

Example #23 Process bands

```
>>> from PIL import Image
>>> source = im.split()
>>> r, g, b = 0, 1, 2
>>> mask = source[r].point(lambda i: i < 100 and 255)
>>> out = source[g].point(lambda i: i * 0.7)
>>> source[g].paste(out, None, mask)
>>> im = Image.merge(im.mode, source)
>>> im.save("masked_hopper.jpg")
```

Example #23 Process bands

```
>>> from PIL import Image
>>> source = im.split()
>>> r, g, b = 0, 1, 2
>>> mask = source[r].point(lambda i: i < 100 and 255)
>>> out = source[g].point(lambda i: i * 0.7)
>>> source[g].paste(out, None, mask)
>>> im = Image.merge(im.mode, source)
>>> im.save("masked_hopper.jpg")
>>> print("Saved masked hopper!")
```

Example #23 Process bands

```
>>> from PIL import Image  
>>> source = im.split()  
>>> r, g, b = 0, 1, 2  
>>> mask = source[r].point(lambda i: i < 100 and 255)  
>>> out = source[g].point(lambda i: i * 0.7)  
>>> source[g].paste(out, None, mask)  
>>> im = Image.merge(im.mode, source)  
>>> im.save("masked_hopper.jpg")  
>>> print("Saved masked hopper!")
```



Example #24 Enhance image

Example #24 Enhance image

```
>>> from PIL import ImageEnhance
```

Example #24 Enhance image

```
>>> from PIL import ImageEnhance  
>>> im = ImageEnhance.Contrast(im)
```

Example #24 Enhance image

```
>>> from PIL import ImageEnhance  
>>> im = ImageEnhance.Contrast(im)  
>>> im = im.enhance(1.3)
```

Example #24 Enhance image

```
>>> from PIL import ImageEnhance  
>>> im = ImageEnhance.Contrast(im)  
>>> im = im.enhance(1.3)  
>>> im.save("contrasted_hopper.jpg")
```

Example #24 Enhance image

```
>>> from PIL import ImageEnhance  
>>> im = ImageEnhance.Contrast(im)  
>>> im = im.enhance(1.3)  
>>> im.save("contrasted_hopper.jpg")  
>>> im.show()
```

Example #24 Enhance image

```
>>> from PIL import ImageEnhance  
>>> im = ImageEnhance.Contrast(im)  
>>> im = im.enhance(1.3)  
>>> im.save("contrasted_hopper.jpg")  
>>> im.show()
```



Example #25 Image sequences

Example #25 Image sequences

```
>>> from PIL import ImageSequence
```

Example #25 Image sequences

```
>>> from PIL import ImageSequence  
>>> im = Image.open("snorkle.gif")
```

Example #25 Image sequences

```
>>> from PIL import ImageSequence  
>>> im = Image.open("snorkle.gif")  
>>> i = 1
```

Example #25 Image sequences

```
>>> from PIL import ImageSequence  
>>> im = Image.open("snorkle.gif")  
>>> i = 1  
>>> for frame in ImageSequence.Iterator(im):
```

Example #25 Image sequences

```
>>> from PIL import ImageSequence  
>>> im = Image.open("snorkle.gif")  
>>> i = 1  
>>> for frame in ImageSequence.Iterator(im):  
>>>     frame.save(f"snorkle_{i}.png")
```

Example #25 Image sequences

```
>>> from PIL import ImageSequence  
>>> im = Image.open("snorkle.gif")  
>>> i = 1  
>>> for frame in ImageSequence.Iterator(im):  
>>>     frame.save(f"snorkle_{i}.png")  
>>>     print(f"Saved snorkle frame {i}!")
```

Example #25 Image sequences

```
>>> from PIL import ImageSequence  
>>> im = Image.open("snorkle.gif")  
>>> i = 1  
>>> for frame in ImageSequence.Iterator(im):  
>>>     frame.save(f"snorkle_{i}.png")  
>>>     print(f"Saved snorkle frame {i}!")  
>>>     i += 1
```

Example #25 Image sequences

```
>>> from PIL import ImageSequence  
>>> im = Image.open("snorkle.gif")  
>>> i = 1  
>>> for frame in ImageSequence.Iterator(im):  
>>>     frame.save(f"snorkle_{i}.png")  
>>>     print(f"Saved snorkle frame {i}!")  
>>>     i += 1
```



Example #26 Postscript

Example #26 Postscript

```
>>> from PIL import PSDraw
```

Example #26 Postscript

```
>>> from PIL import PSDraw  
>>> title = "hopper"
```

Example #26 Postscript

```
>>> from PIL import PSDraw  
>>> title = "hopper"  
>>> fp = open("postscript_hopper.ps", "wb")
```

Example #26 Postscript

```
>>> from PIL import PSDraw  
>>> title = "hopper"  
>>> fp = open("postscript_hopper.ps", "wb")  
>>> ps = PSDraw.PSDraw(fp)
```

Example #26 Postscript

```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
```

Example #26 Postscript

```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
>>> ps.image((0, 0, 128, 128), im, 0)
```

Example #26 Postscript

```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
>>> ps.image((0, 0, 128, 128), im, 0)
>>> ps.setfont("HelveticaNarrow-Bold", 36)
```

Example #26 Postscript

```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
>>> ps.image((0, 0, 128, 128), im, 0)
>>> ps.setfont("HelveticaNarrow-Bold", 36)
>>> ps.text((0, 0), title)
```

Example #26 Postscript

```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
>>> ps.image((0, 0, 128, 128), im, 0)
>>> ps.setfont("HelveticaNarrow-Bold", 36)
>>> ps.text((0, 0), title)
>>> ps.end_document()
```

Example #26 Postscript

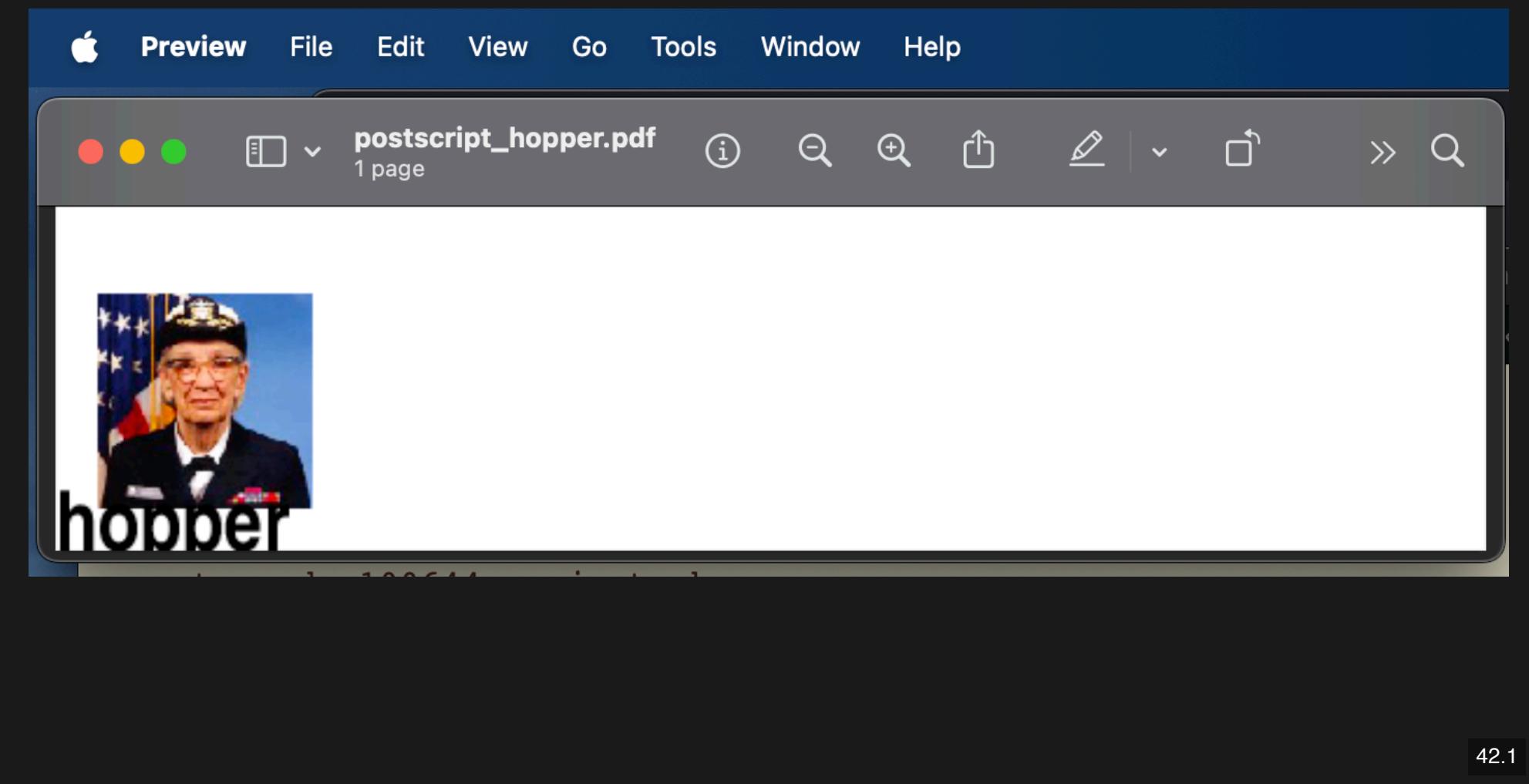
```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
>>> ps.image((0, 0, 128, 128), im, 0)
>>> ps.setfont("HelveticaNarrow-Bold", 36)
>>> ps.text((0, 0), title)
>>> ps.end_document()
>>> print("Saved postscript hopper!")
```

Example #26 Postscript

```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
>>> ps.image((0, 0, 128, 128), im, 0)
>>> ps.setfont("HelveticaNarrow-Bold", 36)
>>> ps.text((0, 0), title)
>>> ps.end_document()
>>> print("Saved postscript hopper!")
convert postscript_hopper.ps postscript_hopper.pdf
```

Example #26 Postscript

Example #26 Postscript



NOT COVERED !!

NOT COVERED !!

Reading from other sources

NOT COVERED !!

Reading from other sources

- <https://pillow.readthedocs.io/en/stable/handbook/tutorial.html#more-on-reading-images>

NOT COVERED !!

NOT COVERED !!

Batch processing

NOT COVERED !!

Batch processing

- <https://pillow.readthedocs.io/en/stable/handbook/tutorial.html#batch-processing>

NOT COVERED !!

NOT COVERED !!

Draft mode

NOT COVERED !!

Draft mode

- <https://pillow.readthedocs.io/en/stable/handbook/tutorial.html#controlling-the-decoder>

NOT COVERED !!

NOT COVERED !!

Logos

NOT COVERED !!

Logos

- <https://github.com/aclark4life/aclarknet-logo>

NOT COVERED !!

Logos

- <https://github.com/aclark4life/aclarknet-logo>
- <https://github.com/aclark4life/dcpython-logo>

NOT COVERED !!

NOT COVERED !!

GIS and VFX

NOT COVERED !!

GIS and VFX

- [https://github.com/python-pillow/Pillow/
issues/1888](https://github.com/python-pillow/Pillow/issues/1888)

NOT COVERED !!

GIS and VFX

- [https://github.com/python-pillow/Pillow/
issues/1888](https://github.com/python-pillow/Pillow/issues/1888)

Probably lots of other stuff...

ALSO CHECK OUT

ALSO CHECK OUT

- <https://pillow.readthedocs.io/en/stable/handbook/concepts.html>

ALSO CHECK OUT

- <https://pillow.readthedocs.io/en/stable/handbook/concepts.html>
- <https://pillow.readthedocs.io/en/stable/handbook/applications.html>

THANK YOU!

THANK YOU!

Alex

Jeffrey A. Clark

Python Pillow Creator

ACLARK.NET, LLC President

DC Python Executive Director

Let's Pay the Maintainers ⚡

aclark@aclark.net

THANK YOU!

Alex

Jeffrey A. Clark

Python Pillow Creator

ACLARK.NET, LLC President

DC Python Executive Director

Let's Pay the Maintainers ⚡

aclark@aclark.net

<https://github.com/python-pillow/pillow-demo>