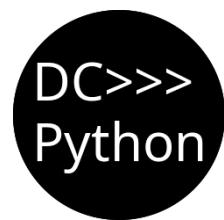




pillow

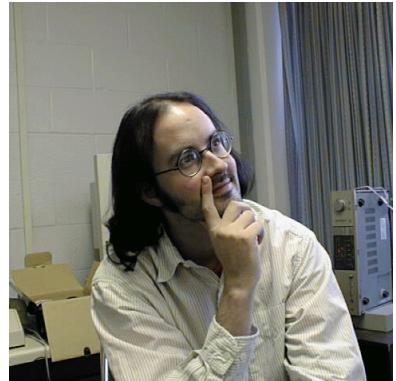


Presented to DC Python Meetup & Aus
Meetup 4/16/24

ABOUT ME



ABOUT ME



Me circa 1998 😊

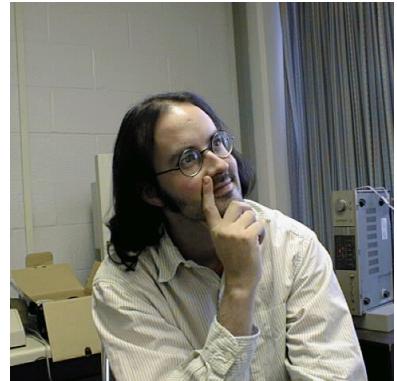
ABOUT ME



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
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 thin

ABOUT ME



Me circa 2008 🤔

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

ABOUT ME



Me circa 2008 🤔

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

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

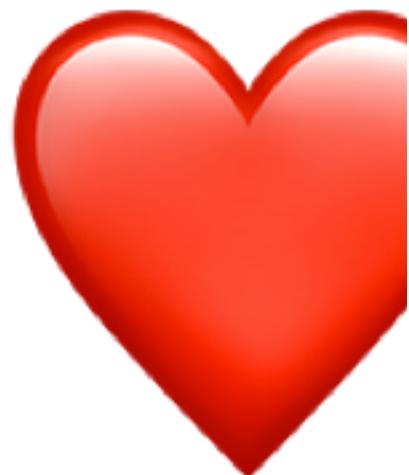
- Lifter 2019 (actually late 2018)
- Lifter Advocate 2023, 2024
- #opentowork (You should probab

TIDELI

TIDELI



TIDELI

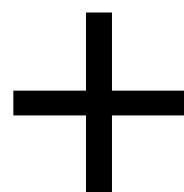


Let's pay the maintainers

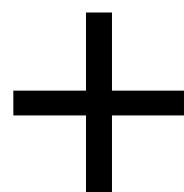


TIDELI

TIDELI



TIDELI



XZ

Let's talk about xz

Let's talk about xz

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

Let's talk about xz

- <https://blog.tidelift.com/xz-tidelift-a-the-maintainers>
- <https://blog.tidelift.com/the-roi-of-p-source-maintainers-in-light-of-xz-ut>

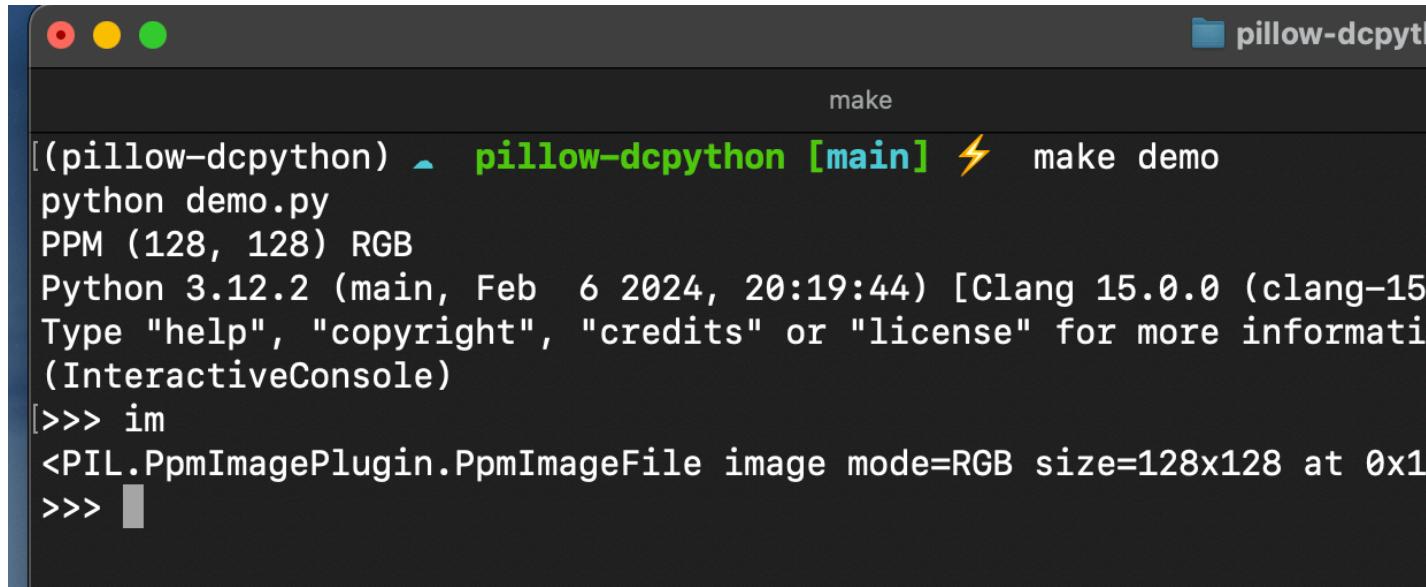
Let's talk about xz

- <https://blog.tidelift.com/xz-tidelift-a-the-maintainers>
- <https://blog.tidelift.com/the-roi-of-p-source-maintainers-in-light-of-xz-utils>
- <https://blog.tidelift.com/maintainer-from-maintainers-in-a-post-xz-utils-world>

Let's talk about Pillow

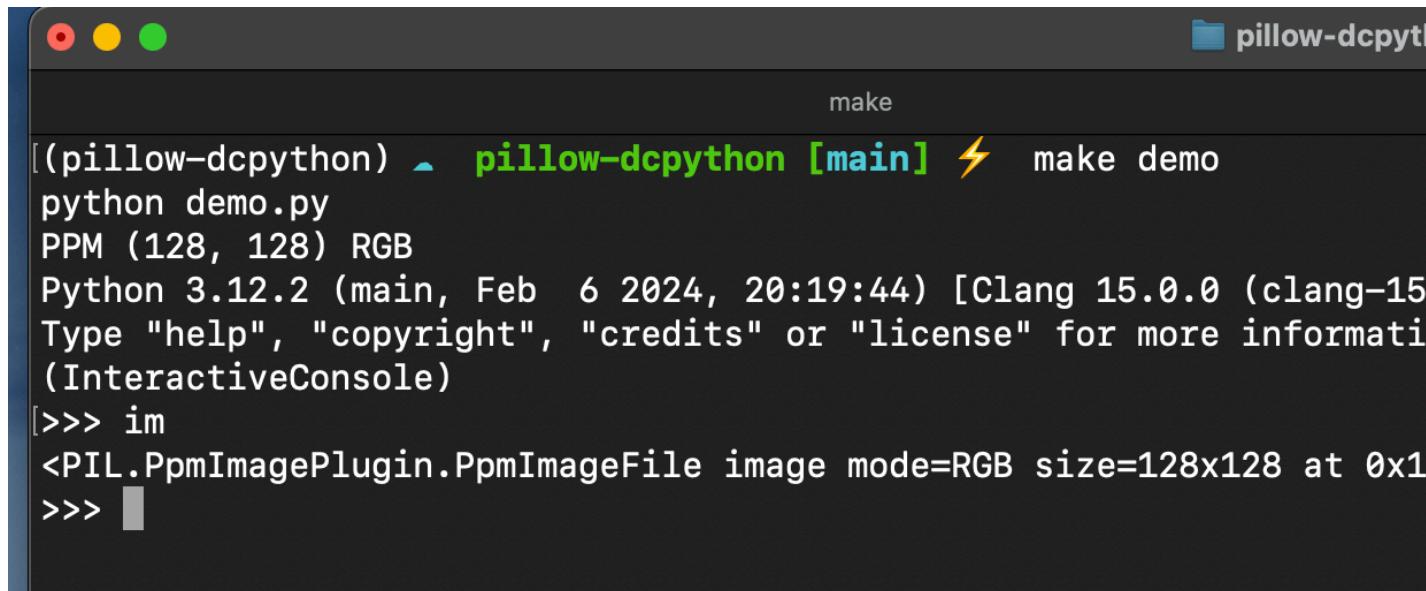
Let's talk about Pillow
(Alex's Version)

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



```
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-1500.0.38)]
Type "help", "copyright", "credits" or "license" for more information
(InteractiveConsole)
[>>> im
<PIL.PpmImagePlugin.PpmImageFile image mode=RGB size=128x128 at 0x1
>>> ]
```

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



A screenshot of a macOS terminal window. The title bar says "pillow-dcpython". The window contains the following text:

```
make
[(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-15
Type "help", "copyright", "credits" or "license" for more information
(InteractiveConsole)
>>> im
<PIL.PpmImagePlugin.PpmImageFile image mode=RGB size=128x128 at 0x1
>>> ]
```

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

Other Examples

Other Examples

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

Other Examples

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

Other Examples

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

Other Examples

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

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

janbodnar / pillow-examples

Type ⌘ to search

Code Issues Pull requests Actions Projects Security Insights

pillow-examples Public

Watch 2 Fork

master 1 Branch 0 Tags Go to file Add file Code

janbodnar Update show_tkinter.py ade4014 · 6 years ago 15 Commits

README.md Update README.md 7 years ago

basic_info.py Create basic_info.py 7 years ago

blur_image.py Create blur_image.py 7 years ago

convert2png.py Create convert2png.py 7 years ago

crop_image.py Create crop_image.py 7 years ago

draw2image.py Create draw2image.py 7 years ago

grayscale.py Create grayscale.py 7 years ago

read_from_url.py Create read_from_url.py 7 years ago

rotate_image.py Create rotate_image.py 7 years ago

show_image.py Create show_image.py 7 years ago

show_tkinter.py Update show_tkinter.py 6 years ago

tatras.jpg Add files via upload 7 years ago

watermark.py Create watermark.py 7 years ago

README

pillow-examples

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

About

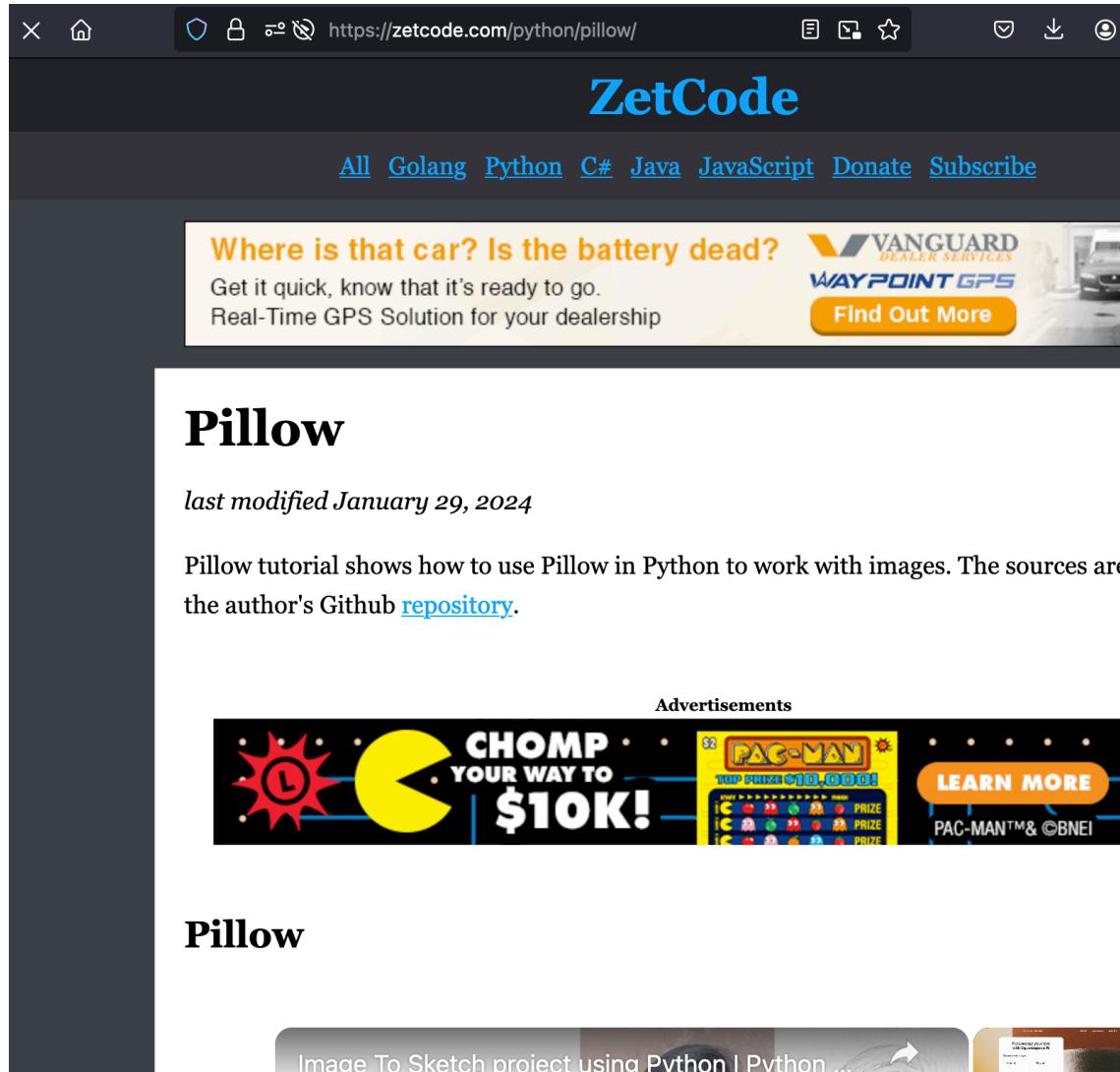
Sources from the Pillow tutorial <http://zetcode.com/python/pillow/>

Readme Activity 2 stars 2 watches 4 forks Report repository

Releases No releases yet

Packages No packages

Languages Python 100%



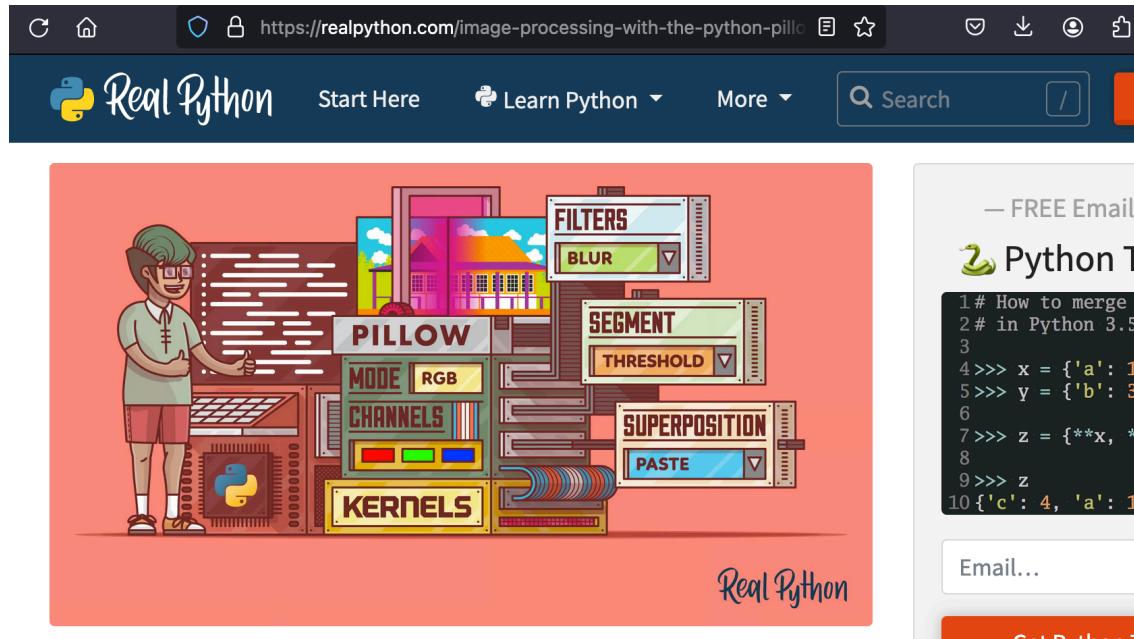


Image Processing With the Python Pillow Library

by Stephen Gruppetta ⌚ Mar 23, 2022 💬 6 Comments

👉 [intermediate](#)

[Mark as Completed](#) Bookmark

[X Share](#) [f Share](#) [Email](#)

Table of Contents

- [Basic Image Operations With the Python Pillow Library](#)

[Browse Topics](#) [Guided Learning Path](#) [Basics](#) [Intermediate](#) [Advanced](#) [api](#) [best-practices](#) [ca](#) [community](#) [databases](#)

The screenshot shows a web browser displaying the Pillow (PIL Fork) documentation handbook at the URL <https://pillow.readthedocs.io/en/stable/handbook/index.html>. The page has a dark theme with white text. At the top left is the Pillow logo, which is a Python icon with a colorful flower. To its right, the word "pillow" is written in a large, lowercase, sans-serif font. Below the logo, the text "Pillow (PIL Fork)" is followed by "10.3.0" and "documentation". A search bar with a magnifying glass icon is positioned below the title. On the left side, there is a navigation sidebar with several sections: "Installation" (with "Overview" and "Tutorial" under it), "Handbook" (which is expanded to show "Concepts", "Appendices", and "Reference" sections), "Porting", "About", "Release Notes" (with "Deprecations and removals" under it), and an "ON-DEMAND PANEL" section. The main content area on the right is titled "Handbook" in a large, bold, white font. Below the title is a bulleted list of topics:

- 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

Example #1 Open image and print

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

Example #1 Open image and print

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

Example #1 Open image and print

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

Example #1 Open image and print

```
>>> 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

```
>>> 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

```
>>> 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
degrees counter clockwise)

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

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

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

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

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

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

Example #2 Rotate image 90 degrees 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

```
>>> 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

```
>>> 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,  
precision 8, 128x128, components 3
```

Example #4 Create thumbnail

Example #4 Create thumbnail

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

Example #4 Create thumbnail

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

Example #4 Create thumbnail

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

Example #4 Create thumbnail

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

Example #4 Create thumbnail

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

Example #4 Create thumbnail

```
>>> 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.R
```

Example #6 Pasting image

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

Example #8 Merge images (and more)

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

Example #8 Merge images (and more)

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

Example #8 Merge images (and more)

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

Example #8 Merge images (and more)

```
>>> 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)

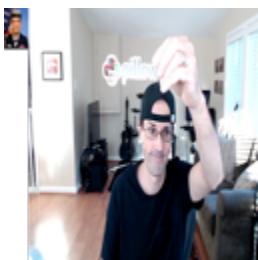
```
>>> from PIL import Image  
>>> hopper = Image.open("hopper.jpg")  
>>> alex = Image.open("img/alex-pillow.jpg")  
>>> im = merge(hopper, alex)  
>>> im.save("merged_resized_hopper.png")  
>>> im = Image.open("merged_resized_hopper.png")  
>>> im = im.resize([128, 128])  
>>> im.save("merged_resized_hopper.png")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

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 bar

Example #9 Split and merge back

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

Example #9 Split and merge back

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

Example #9 Split and merge back

```
>>> 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

Example #10 Create JPEG thumbnail

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

Example #10 Create JPEG thumbnail

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

Example #10 Create JPEG thumbnail

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

Example #10 Create JPEG thumbnail

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

Example #10 Create JPEG thumbnail

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



Example #11 Transpose image left

Example #11 Transpose image left

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

Example #11 Transpose image left

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

Example #11 Transpose image left

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

Example #11 Transpose image left

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

Example #11 Transpose image left

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



Example #12 Transpose image top t

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 w

Example #13 Rotate image 90 degrees w

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

Example #13 Rotate image 90 degrees w

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

Example #13 Rotate image 90 degrees w

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

Example #13 Rotate image 90 degrees w

```
>>> 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 w

```
>>> 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 w

Example #14 Rotate image 180 degrees w

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

Example #14 Rotate image 180 degrees w

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

Example #14 Rotate image 180 degrees w

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

Example #14 Rotate image 180 degrees w

```
>>> 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 w

```
>>> 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 w

Example #15 Rotate image 270 degrees w

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

Example #15 Rotate image 270 degrees

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

Example #15 Rotate image 270 degrees

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

Example #15 Rotate image 270 degrees

```
>>> 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

```
>>> 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

Example #16 Relative resize image with

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

Example #16 Relative resize image with

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

Example #16 Relative resize image with PIL

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

Example #16 Relative resize image with

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

Example #16 Relative resize image with crop

```
>>> 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

```
>>> 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 w

Example #17 Relative resize image w

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

Example #17 Relative resize image w

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

Example #17 Relative resize image w

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

Example #17 Relative resize image w

```
>>> 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 w

```
>>> 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

Example #18 Relative resize image

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

Example #18 Relative resize image

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

Example #18 Relative resize image

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

Example #18 Relative resize image

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

Example #18 Relative resize image

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



Example #19 Relative resize image

Example #19 Relative resize image

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

Example #19 Relative resize image

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

Example #19 Relative resize image

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

Example #19 Relative resize image

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

Example #19 Relative resize image

```
>>> from PIL import ImageOps  
>>> ImageOps.pad(im, (100, 150), color="#f00").show()  
$ 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 operation

Example #22 Point operation

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

Example #22 Point operation

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

Example #22 Point operation

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

Example #22 Point operation

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

Example #22 Point operation

```
>>> 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
```

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  
>>> 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  
>>> 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
>>> 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
>>> 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
>>> 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  
>>> 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 sequence

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

Example #25 Image sequence

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

Example #25 Image sequence

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

Example #25 Image sequence

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

Example #25 Image sequence

```
>>> 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 sequence

```
>>> 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 sequence

```
>>> 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 sequence

```
>>> 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

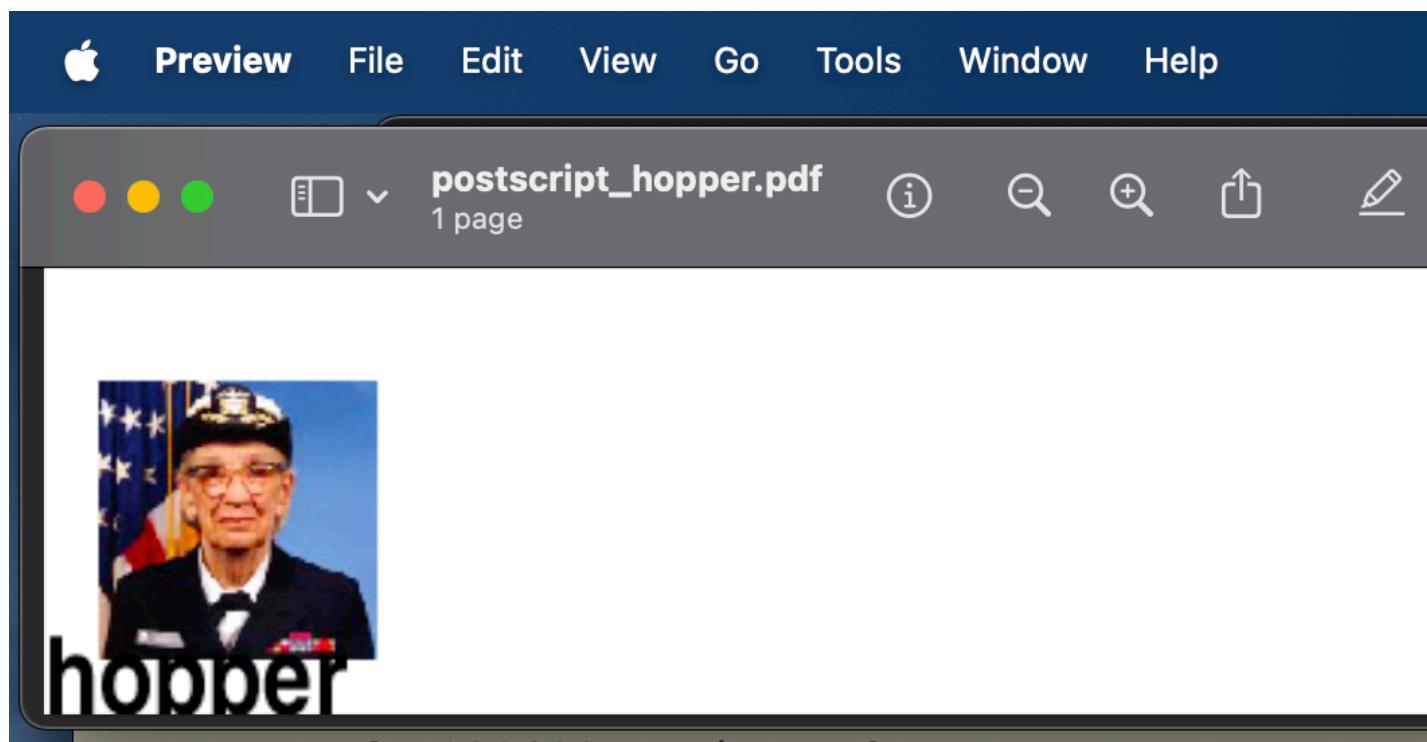
```
>>> from PIL import PSDraw
>>> title = "hopper"
>>> fp = open("postscript_hopper.ps", "wb")
>>> ps = PSDraw.PSDraw(fp)
>>> ps.begin_document(title)
>>> ps.image((0, 0, 128, 128), im, 0)
>>> ps.setfont("HelveticaNarrow-Bold", 36)
>>> ps.text((0, 0), title)
>>> ps.end_document()
>>> print("Saved postscript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

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-references>

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>

NOT COVERED !!

NOT COVERED !!

Logos

NOT COVERED !!

Logos

- <https://github.com/aclark4life/acl>

NOT COVERED !!

Logos

- <https://github.com/aclark4life/aclark4life>
- <https://github.com/aclark4life/dc>

NOT COVERED !!

NOT COVERED !!

GIS and VFX

NOT COVERED !!

GIS and VFX

- <https://github.com/python-pillow/Pissues/1888>

NOT COVERED !!

GIS and VFX

- <https://github.com/python-pillow/Pissues/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/pi>