

pillow その2

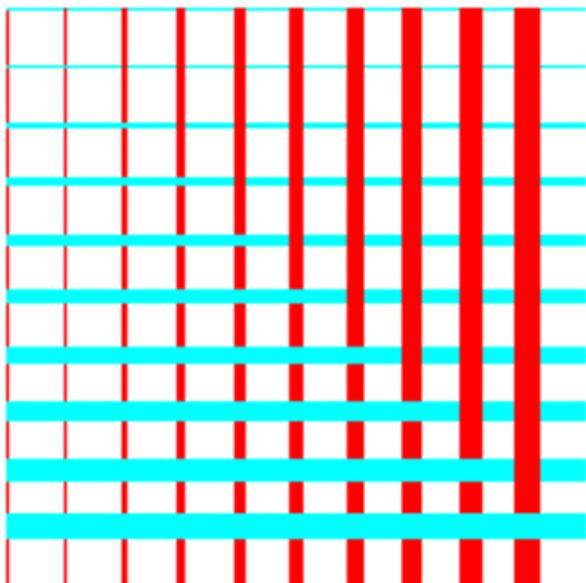
直線描画

[DrawImage].line(位置、fill=色、width=整数)

```
from PIL import Image
from PIL import ImageDraw, ImageColor

img=Image.new('RGB', (300, 300), (255, 255, 255))
draw=ImageDraw.Draw(img)

for n in range(0, 10):
    draw.line([(50+n*20, 50), (50+n*20, 250)], fill=(255, 0, 0), width=n)
    draw.line([(50, 50+n*20), (250, 50+n*20)], fill=(0, 255, 255), width=n)
img
```

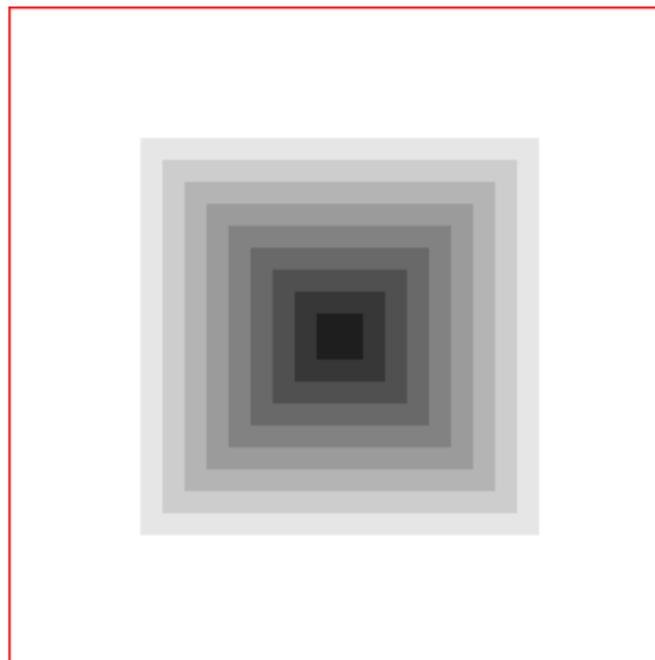


四角形描画

[DrawImage].rectangle(領域、outline=色、fill=色)

```
from PIL import Image
from PIL import ImageDraw, ImageColor

img=Image.new('RGB', (300, 300), (255, 255, 255))
draw=ImageDraw.Draw(img)
draw.rectangle([0, 0, 299, 299], outline=(255, 0, 0))
for n in range(0, 10):
    draw.rectangle([50+n*10, 50+n*10, 250-n*10, 250-n*10], fill=(255-n*25, 255-n*25, 255-n*25))
img
```



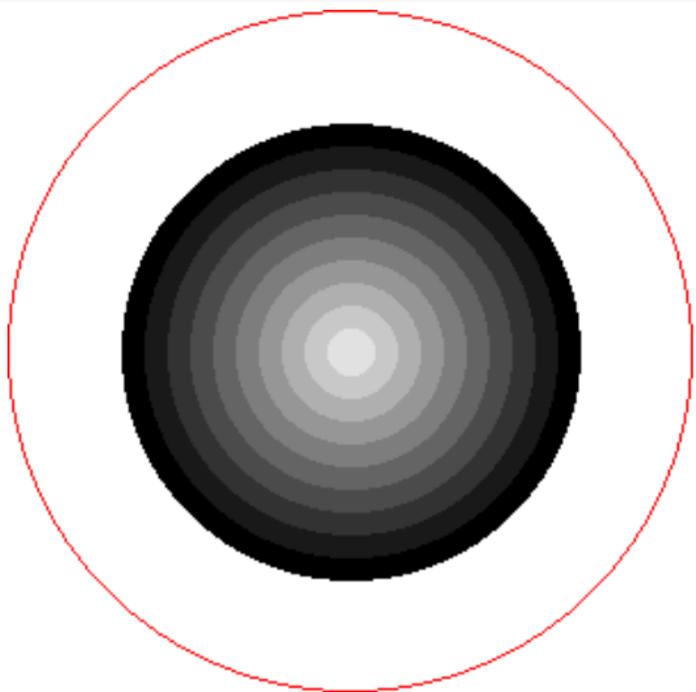
円描画

[DrawImage].ellipse(領域、outline=色、fill=色)

```
from PIL import Image
from PIL import ImageDraw, ImageColor

img=Image.new('RGB', (300, 300), (255, 255, 255))
draw=ImageDraw.Draw(img)
draw.ellipse([0, 0, 299, 299], outline=(255, 0, 0))
for n in range(0, 10):
    draw.ellipse([50+n*10, 50+n*10, 250-n*10, 250-n*10], fill=(n*25, n*25, n*25))

img
```



テキスト描画

[DrawImage].text(位置、テキスト、font=フォント、fill=色)

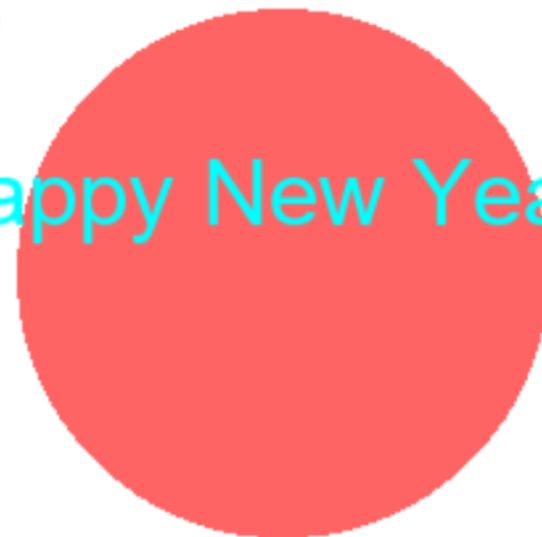
```
from PIL import Image
from PIL import ImageDraw, ImageColor, ImageFont

img=Image.new('RGB', (300, 300), (255, 255, 255))
draw=ImageDraw.Draw(img)
draw.ellipse([50, 50, 250, 250], fill=(255, 100, 100))
fnt=ImageFont.truetype('arial.ttf', 35)
draw.text((10, 50), 'Hello!', fill=(0, 0, 255))
draw.text((10, 100), 'Happy New Year', font=fnt, fill=(0, 255, 255))

img
```

Hello!

Happy New Year



ペースト

[Image].paste([Image]、位置)

```
from PIL import Image
from PIL import ImageDraw, ImageColor, ImageFont

img=Image.open('富士山.jpg')
(w,h)=img.size
img2=img.resize((w//3,h//3))
img.paste(img2, (0,0))
img.paste(img2, (w//3,h//3))
img.paste(img2, (w//3*2,h//3*2))
img
```



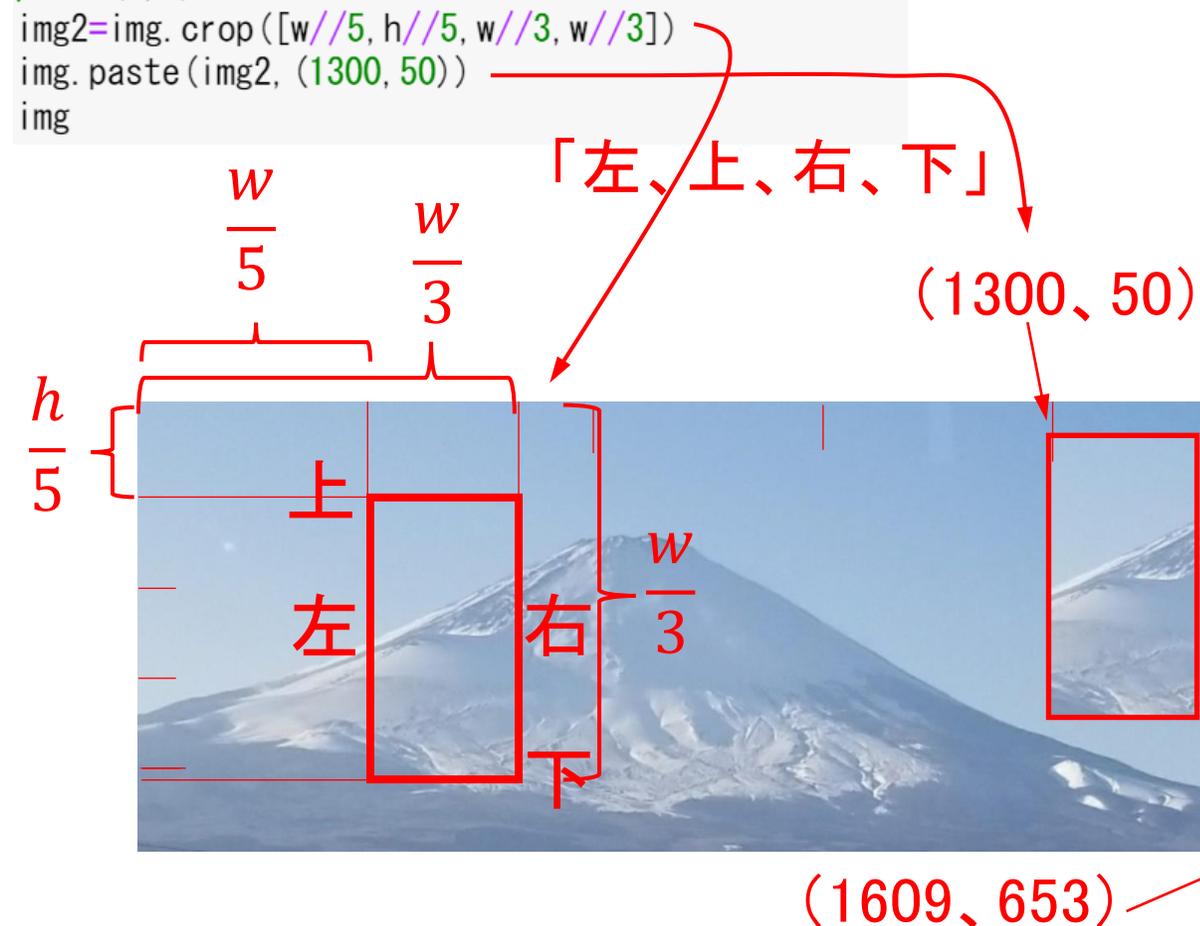
カット

[Image].crop(領域)

```
from PIL import Image
from PIL import ImageDraw, ImageColor, ImageFont

img=Image.open('富士山.jpg')
(w,h)=img.size
print(w,h)
img2=img.crop([w//5,h//5,w//3,w//3])
img.paste(img2, (1300,50))
img
```

1609 653



ブレンド

変数 = Image.blend([Image], [Image], 透過度)

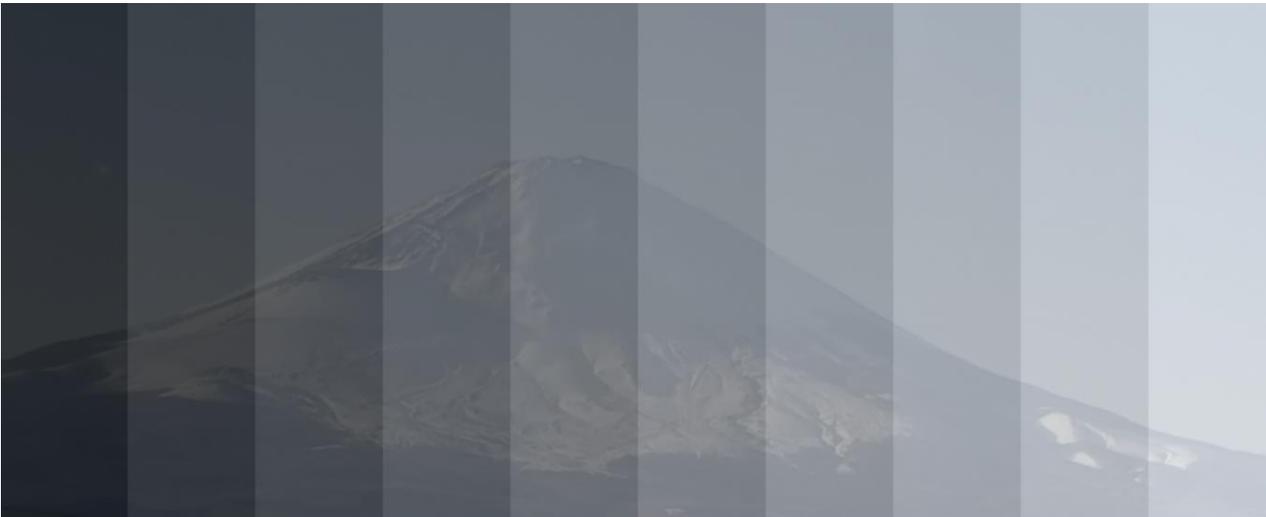
```
from PIL import Image
from PIL import ImageDraw, ImageColor, ImageFont

img=Image.open('富士山.jpg')
(w, h)=img.size
img2=Image.new('RGB', (w, h), (255, 255, 255))
drw=ImageDraw.Draw(img2)
for n in range(0, 10):
    drw.rectangle([w//10*n, 0, w//10*(n+1), h], fill=(n*25, n*25, n*25))
bld_img=Image.blend(img, img2, 0.75)
bld_img
```

セピア色

変数 = [Image].merge(モード、イメージ)

```
import sys
from PIL import Image
img=Image.open('富士山.jpg')
gry=img.convert('L')
r=gry.point(lambda x: x*1.0)
g=gry.point(lambda x: x*0.75)
b=gry.point(lambda x: x*0.5)
img2=Image.merge('RGB', (r, g, b))
img2
```



オートコントラスト

変数=ImageOps.autocontrast([Image]、cutoff=値、ignore=色)

輝度のヒストグラムのcutoff値以下は調整されず
黒くなる

```
from PIL import Image
from PIL import ImageOps

img=Image.open('富士山.jpg')
img2=ImageOps.autocontrast(img,cutoff=25)
img2
```



上下反転 変数=ImageOps.flip([Image])

左右反転 変数=ImageOps.mirror([Image])

輝度反転 変数=ImageOps.invert([Image])

グレースケール化

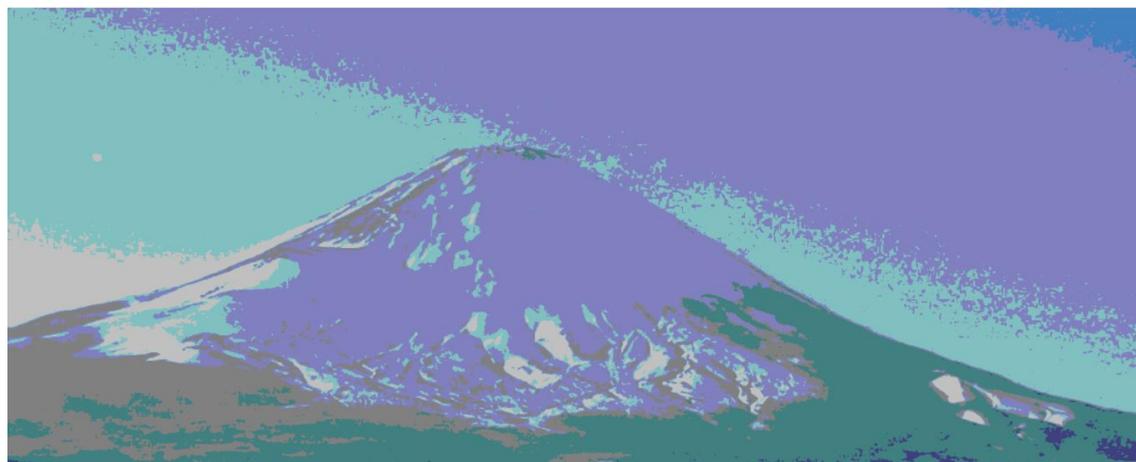
変数=ImageOps.grayscale([Image])

ポスタライズ

変数=ImageOps.posterize([Image]、ビット数)

8ビット(256色) 2ビット(4色) 1ビット(2色)

実行例 2ビット(4色)



ソアリゼーション

変数=ImageOps.solarize([Image]、しきい値)

しきい値: 0~127

```
from PIL import Image
from PIL import ImageOps

img=Image.open('富士山.jpg')
img2=ImageOps.solarize(img, 100)
img2
```



カラーライズ

変数=ImageOps.colorize([Image]、
white=色, black=色)

```
from PIL import Image
from PIL import ImageOps

img=Image.open('富士山.jpg')
img2=ImageOps.colorize(ImageOps.grayscale(img), ¥
                        white=(100,255,255),black=(150,0,0))
img2
```



平均化

変数=ImageOps.equalize([Image])

```
from PIL import Image
from PIL import ImageOps

img=Image.open('富士山.jpg')
img2=ImageOps.equalize(img)
img2
```



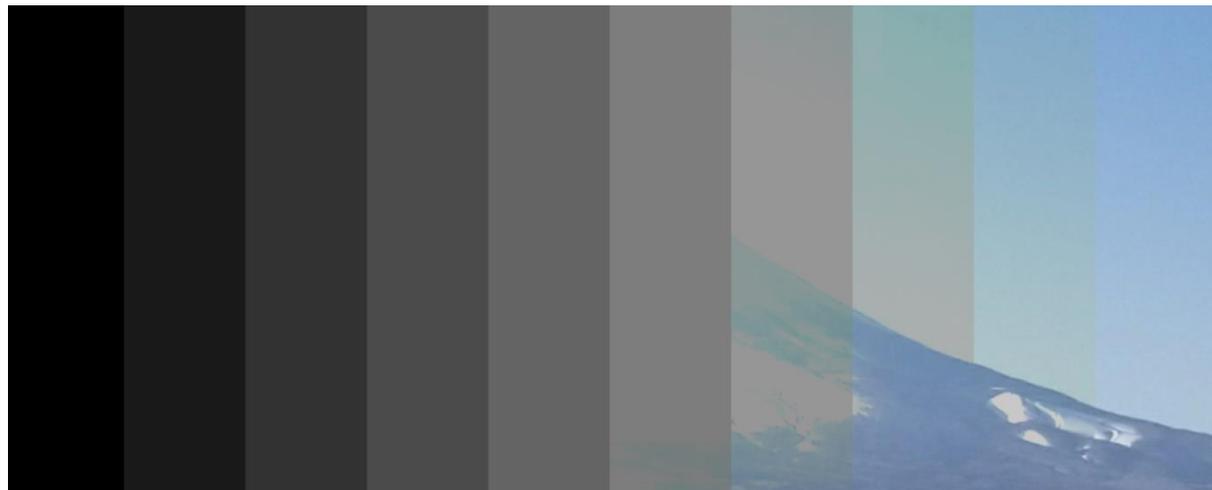
ImageChopsモジュール

darker

2つのイメージの暗い方をとって合成

```
from PIL import Image
from PIL import ImageChops
from PIL import ImageDraw, ImageColor

img=Image.open('富士山.jpg')
(w,h)=img.size
img2=Image.new('RGB', (w,h), (255, 255, 255))
drw=ImageDraw.Draw(img2)
for n in range(0, 10):
    drw.rectangle([w//10*n, 0, w//10*(n+1), h], fill=(n*25, n*25, n*25))
img3=ImageChops.darker(img, img2)
#img3=ImageChops.lighter(img, img2)
img3
```

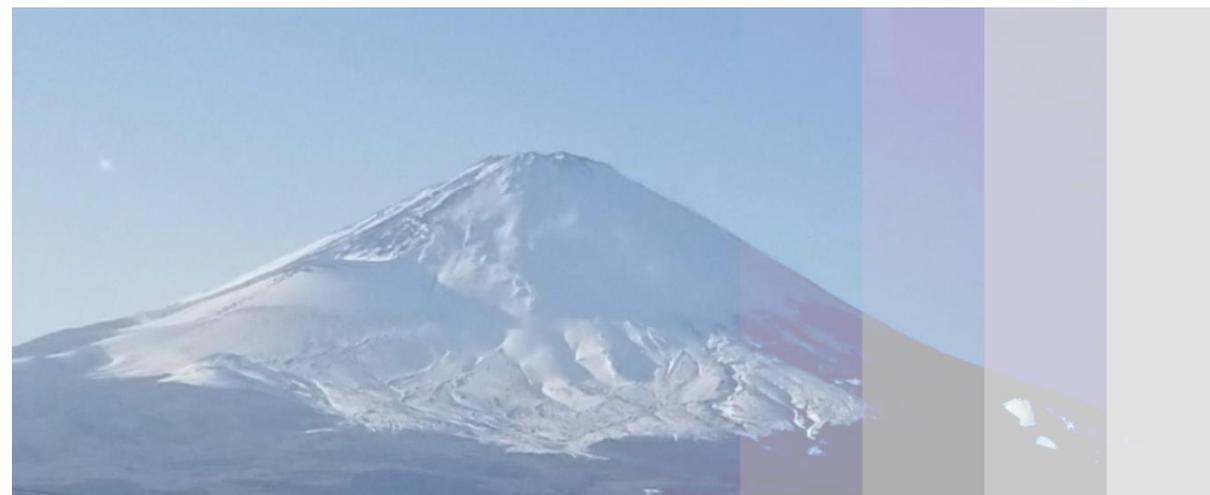


lighter

2つのイメージの明るい方をとって合成

```
from PIL import Image
from PIL import ImageChops
from PIL import ImageDraw, ImageColor

img=Image.open('富士山.jpg')
(w,h)=img.size
img2=Image.new('RGB', (w,h), (255, 255, 255))
drw=ImageDraw.Draw(img2)
for n in range(0, 10):
    drw.rectangle([w//10*n, 0, w//10*(n+1), h], fill=(n*25, n*25, n*25))
#img3=ImageChops.darker(img, img2)
img3=ImageChops.lighter(img, img2)
img3
```



加算 変数=ImageChops.add([Image1], [Image2], scale=実数, offset=整数)

乗算 変数=ImageChops.multiply([Image1], [Image2])

減算 変数=ImageChops.subtract([Image1], [Image2], scale=実数, offset=整数)

差 変数=ImageChops.difference([Image1], [Image2])

マスク 変数=ImageChops.composite([Image1], [Image2], [mask])

```
from PIL import Image
from PIL import ImageChops
from PIL import ImageDraw, ImageColor

img=Image.open('富士山.jpg')
(w,h)=img.size
img2=Image.new('RGB', (w,h), (255,255,255))
drw=ImageDraw.Draw(img2)
for n in range(0,10):
    drw.rectangle([w//10*n, 0, w//10*(n+1), h], fill=(n*25, n*25, n*25))
msk=Image.new('RGB', (w,h), (255,255,255))
msk_drw=ImageDraw.Draw(msk)
msk_drw.ellipse([50,50,w-50,h-50], fill=(0,0,0))
img3=ImageChops.composite(img2, img, msk.convert('L'))
img3
```

Image 1 にmaskで穴をあけ、Image2をはめ込む

グレースケールにマスクで穴をあけ、富士山を貼り付け

