

Python Imaging Library Modules

[[Python Imaging Library](#)][[Python](#)]

Image Module

- <http://www.pythonware.com/library/pil/handbook/image.htm>

new 関数

- モードとサイズを指定してイメージを作成する

```
Image.new(mode, size)
Image.new(mode, size, color)
```

例

```
>>> import Image
>>> im = Image.new('RGB', (100,100), (255,0,0))
>>> im.show()
```



open 関数

- イメージを開く
- ファイル名ではなく、read、seek および tell を持ちバイナリモードで開かれたファイル、string オブジェクトから Image の生成もできる

```
Image.open(infile)
Image.open(infile, mode)
```

ImageDraw Module

- <http://www.pythonware.com/library/pil/handbook/imagenew.htm>

Draw 関数

- イメージに描画するためのインスタンスを取得する

arc

```
draw.arc(xy, start, end, options)
```

- xy で与えられたボックスに、start から、end までの角度の弧を描く。
- options に弧の色を指定。

```
>>> import Image, ImageDraw
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> draw.arc((10,10,80,80), 0, 270, fill=64)
>>> im.save(r'c:\work\img\pil_arc01.jpg')
```

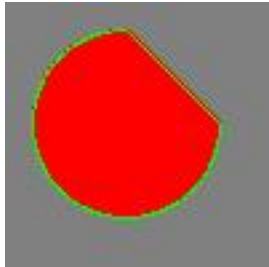


chord

```
draw.chord(xy, start, end, options)
```

- arc と同様だが、エンドポイントが直線で接続されている。
- outline はアウトライン色、fill は塗りつぶす色を指定

```
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> draw.chord((10,10,80,80), 0, 270, outline=(0,255,0), fill=(255,0,0))
>>> im.save(r'c:\work\img\pil_chord01.jpg')
```

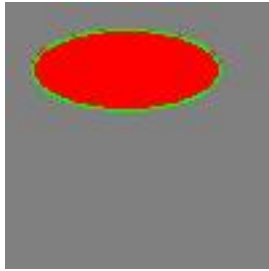


ellipse

```
draw.ellipse(xy, options)
```

- 楕円を指定されたボックス内に描画する。
- outline はアウトライン色、fill は塗りつぶす色を指定

```
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> draw.ellipse((10,10,80,40), outline=(0,255,0), fill=(255,0,0))
>>> im.save(r'c:\work\img\pil_chord01.jpg')
```



line

```
draw.line(xy, options)
```

- xy のリスト間に線を引く
- xy のリストには、シーケンスオブジェクトを指定できる。
 - 2 タプル [(x,y),...] または値のリスト、[x,y, ...]。
- fill オプションで線色を指定できる。
- ver 1.1.5 から、width オプションでピクセル単位で、線幅を指定できる。

```
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> draw.line([(0,0), (50,50), (80,20)], fill='rgb(255,0,0)')
>>> im.save(r'c:\work\img\pil_line01.jpg')
```



pieslice

```
draw.pieslice(xy, start, end, options)
```

- arc と同様だが、エンドポイントと、中心に直線が引かれる。
- outline はアウトライン色、fill は塗りつぶす色を指定

```
>>> import Image, ImageDraw
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> draw.pieslice((10,10,80,80), 0, 270, fill=64)
>>> im.save(r'c:\work\img\pil_slice01.jpg')
```



point

```
draw.point(xy, options)
```

- ・ 点を描画する。
- ・ 2 タプル [(x,y),...] または値のリスト、 [x,y, ...] で位置を指定。

```
>>> import Image, ImageDraw
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> points = [(x, 100.0-(x / 4.0) **2) for x in range(100)]
>>> draw.point(points, fill=64)
>>> im.save(r'c:\work\img\pil_point01.jpg')
```

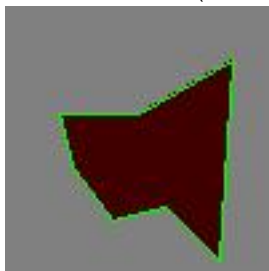


poligon

```
draw.polygon(xy, options)
```

- ・ 多角形を描画する。
- ・ アウトラインを与えられた位置を直線をつないで作成する
- ・ 2 タプル [(x,y),...] または値のリスト、 [x,y, ...] で位置を指定。

```
>>> import Image, ImageDraw
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> points = [(20,40), (25,60), (40,80), (60,75), (80,95), (85,20), (50,40)]
>>> draw.polygon(points, fill=64, outline='rgb(0,255,0)')
>>> im.save(r'c:\work\img\pil_poly01.jpg')
```

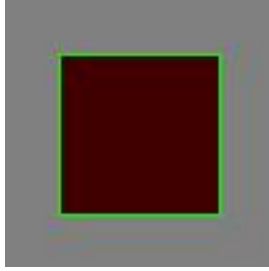


rectangle

```
draw.rectangle(box, options)
```

- ・ 長方形を描画する
- ・ box は、シーケンスオブジェクトで、2 タプル [(x, y), (x, y)] または値 [x, y, x, y] を指定する。

```
>>> import Image, ImageDraw
>>> im = Image.new('RGB', (100,100), 'rgb(128,128,128)')
>>> draw = ImageDraw.Draw(im)
>>> draw.rectangle(((20,20), (80,80))), fill=64, outline='rgb(0,255,0)')
>>> im.save(r'c:\work\img\pil_rectangle01.jpg')
```



text

- ・ 文字列を指定位置に描画する。
- ・ font オプションは、使用するフォントを特定したい場合に利用。
- ・ fill オプションは、テキストの色を指定

フォントを MS 明朝に指定して、日本語を書き出す

```
# -*- coding: utf-8 -*-  
  
import Image, ImageDraw, ImageFont  
im = Image.new('RGB', (300,300), 'rgb(255,255,255)')  
draw = ImageDraw.Draw(im)  
font = ImageFont.truetype('C:\WINDOWS\Fonts\MSMINCHO.TTC', 24, encoding="utf-8")  
draw.text((10,10), u'日本語 OK ? ', font=font, fill='rgb(255,0,0)')  
im.save(r'c:\work\img\pil_text01.jpg')
```

日本語 OK ?

textsize

```
draw.textsize(string, options) => (width, height)
```

- ・ 与えられた文字列のサイズをピクセル単位で取得する。

```
# -*- coding: utf-8 -*-  
  
import Image, ImageDraw, ImageFont  
im = Image.new('RGB', (300,100), 'rgb(255,255,255)')  
draw = ImageDraw.Draw(im)  
font = ImageFont.truetype('C:\WINDOWS\Fonts\MSMINCHO.TTC', 24, encoding="utf-8")  
msg = u'日本語 OK ?'  
(w, h) = draw.textsize(msg, font=font)  
top = 10  
draw.text((top,10), msg, font=font, fill='rgb(255,0,0)')  
top += (h + 2)  
msg = 'w:%d h:%d' % (w, h)  
draw.text((10,top), msg, font=font, fill='rgb(255,0,0)')  
im.save(r'c:\work\img\pil_text02.jpg')
```

日本語OK？
w:144 h:24