



Pillow

Live de Python # 176



Roteiro

1. Pillow

Sobre a biblioteca e funções básicas

2. O que é uma Imagem?

Um pouco de teoria sobre o que vimos até agora

3. Realces

Embelezando fotos de beijos

4. Textos e outra firulas

Inserindo textos, adicionando bordas, etc.



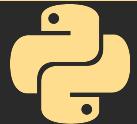
picpay.me/dunossauro



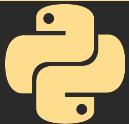
apoia.se/livedepython



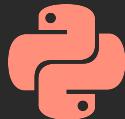
PIX



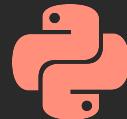
Ajude o projeto



Ademar Peixoto, Alex Lima, Alexandre Harano, Alexandre Santos, Alexandre Tsuno, Alexandre Villares, Alynne Ferreira, Alysson Oliveira, Amaziles Carvalho, André Rocha, Arnaldo Turque, Artur Zalewska, Bruno Batista, Bruno Oliveira, Caio Nascimento, Carlos Chiarelli, Cleber Santos, César Almeida, Davi Ramos, David Kwast, Diego Guimarães, Diego Ubirajara, Dilenon Delfino, Donivaldo Sarzi, Elias Soares, Emerson Rafael, Eric Niens, Érico Andrei, Eugenio Mazzini, Everton Alves, Fabiano Gomes, Fabio Barros, Fabio Castro, Flavkaze Flavkaze, Francisco Alencar, Franklin Silva, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Castro, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Henrique Machado, Israel Fabiano, Israel Gomes, Italo Silva, Jair Andrade, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jorge Plautz, Jose Mazolini, José Prado, João Lugão, Juan Gutierrez, Jônatas Silva, Kaio Peixoto, Kaneson Alves, Leonardo Cruz, Leonardo Mello, Leonardo Nascimento, Lidiane Monteiro, Lorena Ribeiro, Lucas Barros, Lucas Mello, Lucas Mendes, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Luciano Silva, Maiquel Leonel, Marcela Campos, Maria Clara, Marina Passos, Matheus Vian, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Patric Lacouth, Patricia Minamizawa, Patrick Brito, Patrick Gomes, Paulo Tadei, Pedro Pereira, Peterson Santos, Rafael Lino, Reinaldo Silva, Renan Moura, Revton Silva, Rodrigo Ferreira, Rodrigo Mende, Rodrigo Vaccari, Ronaldo Silva, Sandro Mio, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Moraes, Tyrone Damasceno, Vinícius Bastos, Vítor Gomes, Wendel Rios, Wesley Mendes, Willian Lopes, Willian Rosa, Wilson Duarte, Yury Barros



Obrigado você



Sobre a biblioteca
e suas funções
básicas

Pillow

Pillow



Pillow é uma biblioteca para manipulação de imagens com Python.

É fork de uma grande biblioteca, que infelizmente nos deixou no Python 2, a **PIL** (Python Image Library) e atualizada constantemente

Versão atual: 8.3.2

Primeira versão: 1995 (PIL), 2009 (Pillow)

Licença:

pip install pillow



Instalação





Passos fundamentais

```
1  """Exemplo 01."""
2  from PIL import Image
3
4  # Abre a imagem
5  im = Image.open('beijo.jpg')
6
7  # Mostra a imagem
8  im.show( 'Minha imagem!' )
9
10 # Salva a imagem
11 im.save( 'exemplo.jpg' )
```



Passos fundamentais

```
1 """Exemplo 01."""
2 from PIL import Image
3
4 # Abre a imagem
5 im = Image.open('beijo.jpg')
6
7 # Mostra a imagem
8 im.show('Minha imagem!')
9
10 # Salva a imagem
11 im.save('exemplo.jpg')
```



beijo.jpg



exemplo.jpg



Passos fundamentais - Conversão de tipo

```
1 """Exemplo 02."""
2 from PIL import Image
3
4 # Abre a imagem
5 im = Image.open('beijo.jpg')
6
7 # Mostra a imagem
8 im.show('Minha imagem!')
9
10 # Salva a imagem em png
11 im.save('exemplo.png')
```



beijo.jpg



exemplo.png



Passos fundamentais - Redimensionar imagem

```
1  """Exemplo 03."""
2  from PIL import Image
3
4
5  im = Image.open('beijo.jpg') ←
6
7  tamanho = (1280, 720) # x, y
8
9  # Gera um novo objeto Image
10 im_menor = im.resize(tamanho)
11
12 im_menor.show()
13
14 im_menor.save('beijo_menor.jpg')
```



beijo.jpg



beijo_menor.png



Passos fundamentais - Rotação da imagem

```
1 """Exemplo 04."""
2 from PIL import Image
3
4 im = Image.open('beijo_menor.jpg') ←
5
6
7 angulo = 45
8 im.rotate(angulo).show() →
9 im.rotate(angulo, expand=True).show()
```



beijo_menor.jpg



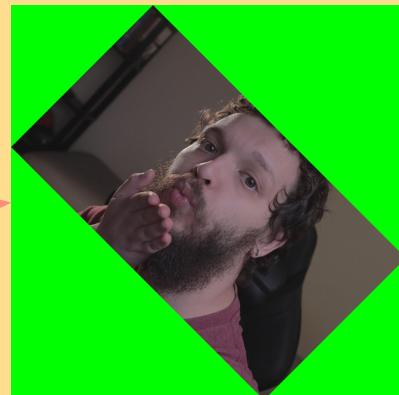


Passos fundamentais - Rotação da imagem

```
1 """Exemplo 04."""
2 from PIL import Image
3
4 im = Image.open('beijo_menor.jpg') ←
5
6
7 angulo = 45
8 im.rotate(angulo).show()
9 im.rotate(angulo, expand=True).show() →
```



beijo_menor.jpg





```
1  """Exemplo 05."""
2  from PIL import Image
3
4  # Abre a imagem
5  im = Image.open('beijo_menor.jpg')
6
7  # Flip
8  # Inverte na horizontal
9  im.transpose(Image.FLIP_LEFT_RIGHT).show()
10 # Inverte na vertical
11 im.transpose(Image.FLIP_TOP_BOTTOM).show()
12
13 # Transposição
14 im.transpose(Image.ROTATE_90).show()
15 im.transpose(Image.ROTATE_180).show()
16 im.transpose(Image.ROTATE_270).show()
17 im.transpose(Image.TRANSPOSE).show()
18 im.transpose(Image.TRANSVERSE).show()
```



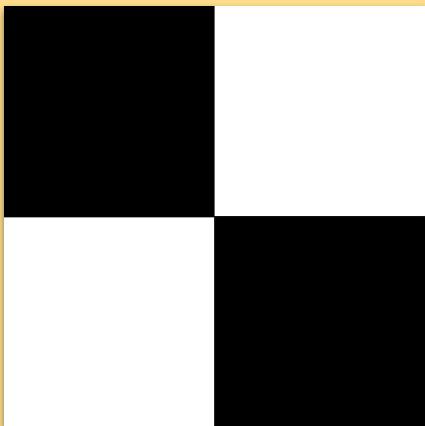
```
1     """Exemplo 05."""
2     from PIL import Image
3
4     # Abre a imagem
5     im = Image.open('beijo_menor.jpg')
6
7     # Flip
8     # Invete na horizontal
9     im.transpose(Image.FLIP_LEFT_RIGHT).show()
10    # Invete na vertical
11    im.transpose(Image.FLIP_TOP_BOTTOM).show()
12
13    # Transposição
14    im.transpose(Image.ROTATE_90).show()
15    im.transpose(Image.ROTATE_180).show()
16    im.transpose(Image.ROTATE_270).show()
17    im.transpose(Image.TRANSPOSE).show()
18    im.transpose(Image.TRANSVERSE).show()
```

O que é uma imagem?

Um pouco de
teoria, mas não
muito



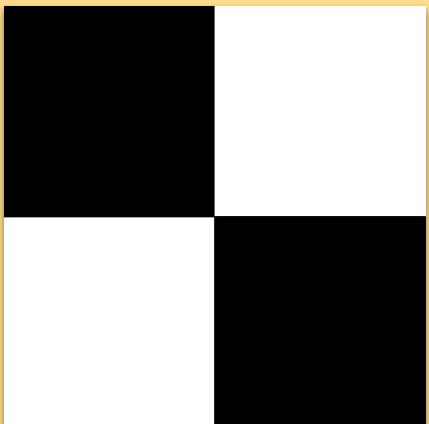
Uma imagem



2x2 Pixels



Uma imagem



2x2 Pixels

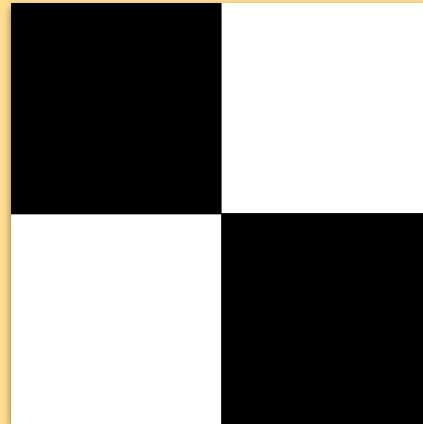
Tá, mas como o
computador
entende isso?



Uma imagem

– □ ×

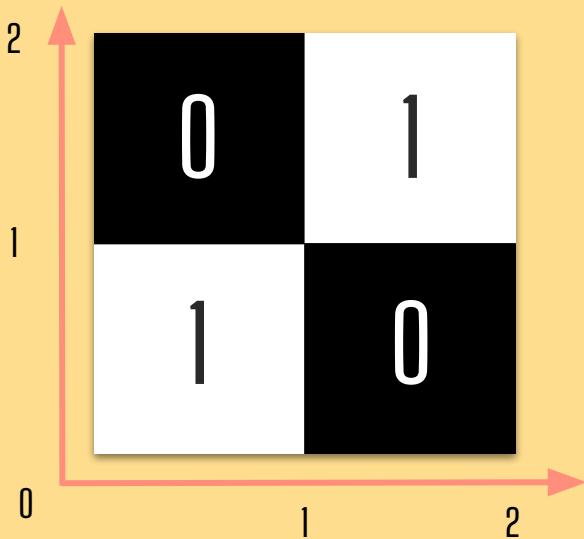
```
1  """Exemplo 06."""
2  from PIL import Image
3
4  im = Image.open('2x2px.jpg')
5
6  print(im.size) # (2, 2)
7  print(im.mode) # 'RGB'
8  print(im.bits) # 8
9
10 print(2**8)    # 256
```



2x2 Pixels



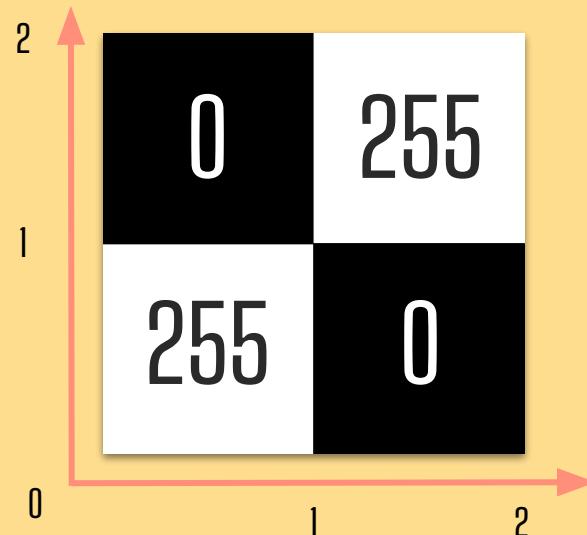
Uma imagem





Uma imagem

```
1  """Exemplo 06."""
2  from PIL import Image
3
4  im = Image.open('2x2px.jpg')
5
6  print(im.size) # (2, 2)
7  print(im.mode) # 'RGB'
8  print(im.bits) # 8
9
10 print(2**8)    # 256
```





Uma matriz

$$imagem = \begin{bmatrix} 0 & 255 \\ 255 & 0 \end{bmatrix}$$

```
1 """Exemplo 07."""
2 from PIL import Image
3 from numpy import array
4
5 im = Image.open('2x2px.jpg')
6
7 >>> print(array(im))
8 array([[[ 0,  0,  0],
9         [255, 255, 255]],
10        [[255, 255, 255],
11         [ 0,  0,  0]]], dtype=uint8)
```

Rotacionando



$$\begin{bmatrix} 0 & 255 \\ 255 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 255 & 0 \\ 0 & 255 \end{bmatrix}$$

Rotacionando

```
1  """Exemplo 08."""
2  from PIL import Image
3  from numpy import array, rot90
4
5  im = Image.open('2x2px.jpg')
6
7  Image.fromarray( # Nova imagem
8      rot90( # Rotaciona a matriz em 90
9          array(im) # matriz
10         )
11     ).show()
```



$$\begin{bmatrix} 255 & 0 \\ 0 & 255 \end{bmatrix}$$



Uma introdução aos modos

| Modo | O que é? |
|------|---|
| 1 | O pixel só pode ter 2 valores, Verdadeiro ou Falso |
| L | L - 256 valores em preto e branco (Escala de cinza) |
| P | P - 256 valores coloridos |
| RGB | RGB - 256 em cada camada ("padrão") |



Modos

L



1



RGB



P





Olhando as matrizes de modos

- □ ×

```
1  """Exemplo 09."""
2  from PIL import Image
3  from numpy import array
4
5  im = Image.open('2x2px.jpg')
6
7  array(im.convert('1')) # True / False
8  array(im.convert('L')) # 0 - 255
9  array(im.convert('P')) # 0 - 255
10 array(im.convert('RGB'))
```

Filtros e Realces

Embelezando
beijos



Realce

A grande sacada da classe de melhorias é fazer ajustes de:

- Balanço de cor
- Contrace
- Brilho
- Sombras



Contraste

- □ ×

```
1 """Exemplo 10."""
2 from PIL import Image, ImageEnhance
3
4 im = Image.open('beijo.jpg')
5
6 contrast = ImageEnhance.Contrast(im)
7 im_contrast = contrast.enhance(1.8)
8
9 im_contrast.show()
10
```





Usando todas os tratamentos

- □ ×

```
1 """Exemplo 11."""
2 from PIL import Image, ImageEnhance
3
4 image = Image.open('beijo_menor.jpg')
5
6 saturation = ImageEnhance.Color(image)
7 contrast = ImageEnhance.Contrast(image)
8 brightness = ImageEnhance.Brightness(image)
9 sharpness = ImageEnhance.Sharpness(image)
```



Meu filtro de Instagram

```
1  """Exemplo 11."""
2  def duno_filtro(img):
3      contrast = ImageEnhance.Contrast(img)
4      contrastado = contrast.enhance(1.1)
5
6      brightness = ImageEnhance.Brightness(contrastado)
7      brilho = brightness.enhance(1.1)
8
9      saturation = ImageEnhance.Color(brilho)
10     saturada = saturation.enhance(1.3)
11
12     return saturada
13
14
15 duno_filtro(image).save('foto_com_filtro.jpg')
```



PERA,
tá faltando algo



Uma vinheta!





Roubando com a vinheta

```
1 """Exemplo 12."""
2
3 def duno_filtro(img):
4     ...
5
6     # Colando outra imagem
7     vinheta = Image.open('vinheta.png')
8     saturada.paste(vinheta, (0, 0), vinheta)
9
10    saturada.save('beijo_editada.jpg')
11
12
13 duno_filtro(image)
```



Textos

E coisas mais



Escrevendo em uma imagem

```
1 """Exemplo 13."""
2 from PIL import Image, ImageDraw
3
4 im = Image.new( # Cria uma nova imagem
5     'RGB', (200, 200), 'green'
6 )
7
8 draw = ImageDraw.Draw(im)
9 draw.text((10, 10), 'Live de Python! ')
10
11
12 im.show('imagem_com_texto.jpg')
13
```

Live de Python



Escrevendo com a sua fonte

```
1 """Exemplo 14."""
2 from PIL import Image, ImageDraw, ImageFont
3
4 im = Image.open('beijo_menor.jpg')
5
6 draw = ImageDraw.Draw(im)
7 font = ImageFont.truetype(
8     'Abys-Regular.otf',
9     size=150
10 )
11
12 draw.text((10, 10), 'Live de Python!', font=font)
13
14
15 im.show('imagem_com_texto.jpg')
```



Bora escrever um certificado



Bora!!





Adicionando bordas

```
1 """Exemplo 15."""
2 from PIL import Image, ImageOps
3
4 img = Image.open('beijo_menor.jpg')
5
6 new_img = ImageOps.expand(
7     img,
8     border=30,
9     fill='white'
10 )
11
12 new_img.show()
```



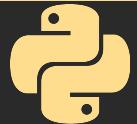
picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto

