

Sugestão para aprender Python para quem já conhece C/C++:

http://www.aleax.it/goo_py4prog.pdf

O programa abaixo lê uma imagem e mostra na tela. Em Linux, só funciona com Python2 e OpenCV2. No OpenCV3 ou OpenCV4, dá erro no imshow.

```
#leimg1.py: So funciona no python2/OpenCV2
```

```
import cv2;
import sys;

nome="lenna.jpg";
a=cv2.imread(nome,1);
cv2.imshow("janela",a);
cv2.waitKey();
```

Para contornar este problema, dá para usar função que mostra imagem do matplotlib. O programa abaixo funciona no Python 2 e 3, OpenCV 2, 3 e 4.

```
#leimg2.py
```

```
#Mostra na tela usando funcao do matplotlib
```

```
import cv2;
import sys;
from matplotlib import pyplot as plt
```

```
nome="lenna.jpg";
a=cv2.imread(nome,1);
a=cv2.cvtColor(a,cv2.COLOR_BGR2RGB);
plt.imshow(a,interpolation="bicubic")
plt.show()
```

Matplotlib usa convenção RGB. OpenCV usa convenção BGR. Assim é necessário converter um sistema de cor para outro. Para executar:

```
>python leimg2.py
$python3 leimg2.py
```

O programa abaixo reclama se não achar o arquivo:

```
#leimg3.py
```

```
#Mostra mensagem de erro se nao achar imagem
```

```
import cv2;
import sys;
from matplotlib import pyplot as plt
```

```
nome="lennax.jpg";
a=cv2.imread(nome,1);
if (a is None) or (a.size==0):
    print("Nao achei imagem",nome);
    sys.exit();

a=cv2.cvtColor(a,cv2.COLOR_BGR2RGB);
plt.imshow(a,interpolation="bicubic");
plt.show();
```

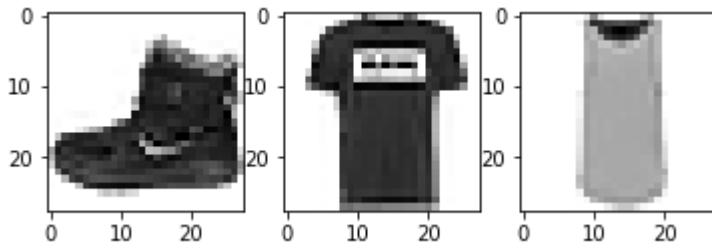
Mostrar imagem em níveis de cinza:

```
#leimg4.py
#Mostra imagem em niveis de cinzas na tela usando matplotlib
#Faz interpolacao bicubica.
import cv2;
import sys;
from matplotlib import pyplot as plt

nome="lenna.jpg";
a=cv2.imread(nome,0);
plt.imshow(a,cmap="gray",interpolation="bicubic");
plt.show();
```

Mostrar 3 imagens em níveis de cinza lado a lado:

```
from matplotlib import pyplot as plt
f = plt.figure()
f.add_subplot(1,3,1)
plt.imshow(img1,cmap="gray")
f.add_subplot(1,3,2)
plt.imshow(img2,cmap="gray")
f.add_subplot(1,3,3)
plt.imshow(img3,cmap="gray")
plt.show(block=True)
```



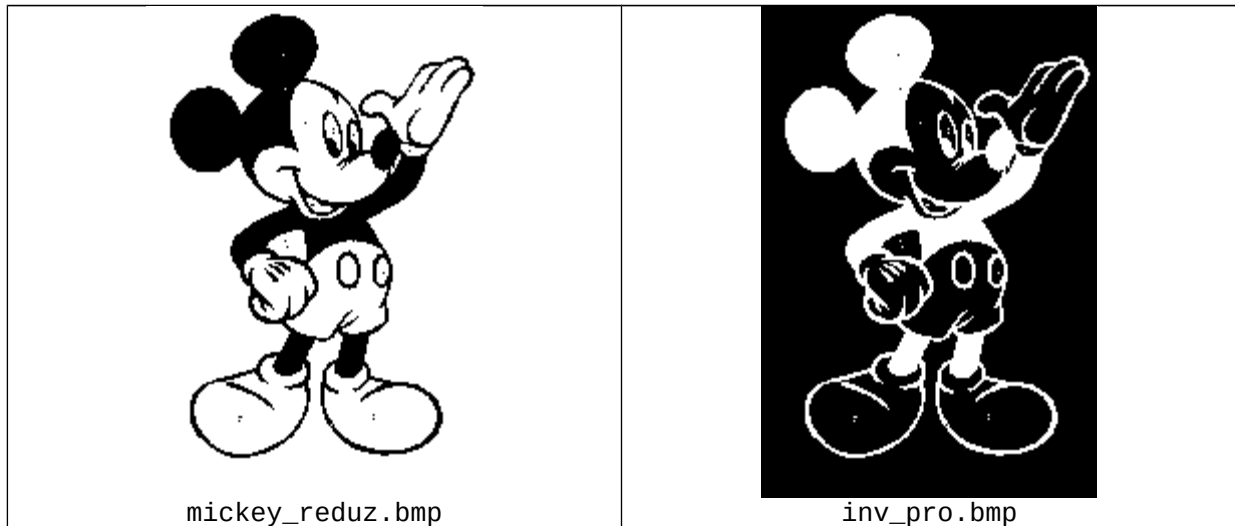
Programa abaixo imprime o tipo de variável *a*, o tipo de cada elemento da matriz *a*, número de dimensões de *a* e o forma de *a*:

```
import numpy as np
a=np.full((100,100,3),(255,0,0),dtype="uint8")
print(type(a),a.dtype,a.ndim,a.shape)
```

```
<class 'numpy.ndarray'> uint8 3 (100, 100, 3)
```

Lê uma imagem binária e inverte a cor dos pixels.

Nota: A numeração das linhas vai de 0 até a.rows-1, de cima para baixo. A numeração das colunas vai de 0 até a.cols-1, da esquerda para direita.



```
# inv_cek.py
import cv2
a = cv2.imread('mickey_reduz.bmp', 0)
#print a.shape[0]

for l in range(a.shape[0]):
    for c in range(a.shape[1]):
        if a[l,c]==0:
            a[l,c]=255
        else:
            a[l,c]=0

cv2.imwrite('inv_cek_py.png', a)

hae@Royale ~/haepi/opencv-py $ time inv_cek
real    0m0.060s
hae@Royale ~/haepi/opencv-py $ time python inv_cek.py
real    0m0.432s
```

Em Python, ficou 7 vezes mais lento que C++.

Neste caso, dá para escrever o mesmo programa sem loop:

```
# inv_cek.py
import cv2
a = cv2.imread('mickey_reduz.bmp', 0)
a = 255-a
cv2.imwrite('inv_cek_py.png', a)
```

Os tutoriais de OpenCV-Python estão em ("tutroals" mesmo):
https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html

//invertec.cpp: Inverte as cores da imagem colorida



```
#invertec.py
import cv2
a=cv2.imread("lenna.jpg",1)
for l in range(a.shape[0]):
    for c in range(a.shape[1]):
        a[l,c,0]=255-a[l,c,0]
        a[l,c,1]=255-a[l,c,1]
        a[l,c,2]=255-a[l,c,2]
cv2.imwrite("invertec_py.jpg",a);
```

Aqui também dá para escrever sem loop:

```
#invertec.py
import cv2
a=cv2.imread("lenna.jpg",1)
a=255-a
cv2.imwrite("invertec_py.jpg",a);
```

Exemplos de manipulação de imagens em OpenCV_Python:

```
#Matriz.py
#Manipulacao basica de imagens em Opencv_Python
import cv2
import numpy as np

#Leitura de imagem colorida
a=cv2.imread("lenna.jpg",1)
print(a.shape)
print(a.dtype)

#ROI
roi=a[0:256,0:512]
print(roi.shape)
cv2.imshow('janela',roi)
cv2.waitKey(0)

#Conversao para grayscale
b=cv2.cvtColor(a,code=cv2.COLOR_BGR2GRAY)
cv2.imshow('janela',b)
cv2.waitKey(0)

#Leitura de imagem grayscale
a=cv2.imread("lenna.jpg",0)
print(a.shape)
print(a.dtype)

#Conversao de tipo de matriz
f = np.float32(a)
f = f/255.0
print(f.shape)
print(f.dtype)

#Criacao de matriz vazia
b=np.empty((512,512,3),dtype=np.uint8)
print(b.shape)
print(b.dtype)

'''
$ python matriz.py
(512, 512, 3)
uint8
(256, 512, 3)
(512, 512)
uint8
(512, 512)
float32
(512, 512, 3)
uint8
'''
```

```
#testacor.py
import cv2
import numpy as np
import cek

a=cv2.imread("lenna.jpg")
for l in range(50):
    for c in range(50):
        a[l,c,:]=(0,0,255)

print( a[0,0,:]=(0,0,255) )

#if np.all( a[0,0,:]=(0,0,255) ):
if np.array_equal( a[0,0,:], (0,0,255) ):
    print("vermelho")
else:
    print("nao-vermelho")

cek.mostra(a)
```

Cria imagem colorida e preenche com uma certa cor:

```
img=np.full( (480,640,3), (0,0,255), dtype="uint8")
```