

# Introdução: Open CV

```
In [1]: import numpy as np
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: exemplo = cv2.imread('static/exemplo1.jpg') #carregar uma imagem com o Open CV
```

## 1 - A imagem já é carregada como matriz

```
In [3]: exemplo
```

```

Out[3]: array([[142, 105, 67],
               [143, 106, 68],
               [143, 106, 68],
               ...,
               [145, 121, 85],
               [145, 121, 85],
               [145, 121, 85]],

          [[140, 103, 65],
           [142, 105, 67],
           [142, 105, 67],
           ...,
           [150, 126, 90],
           [150, 126, 90],
           [150, 126, 90]],

          [[139, 102, 64],
           [141, 104, 66],
           [141, 105, 65],
           ...,
           [151, 127, 91],
           [151, 127, 91],
           [151, 127, 91]],

          ...,

          [[ 40,  52,  24],
           [ 44,  56,  28],
           [ 47,  57,  34],
           ...,
           [ 75, 111,  64],
           [ 76, 110,  63],
           [ 73, 107,  60]],

          [[ 38,  50,  22],
           [ 41,  53,  25],
           [ 45,  55,  32],
           ...,
           [ 75, 111,  64],
           [ 75, 109,  62],
           [ 72, 106,  59]],

          [[ 35,  47,  19],
           [ 39,  51,  23],
           [ 42,  52,  29],
           ...,
           [ 78, 114,  67],
           [ 77, 111,  64],
           [ 73, 107,  60]]], dtype=uint8)

```

## 2 - O Open CV usa o padrão BGR

```
In [4]: plt.imshow(exemplo)
```

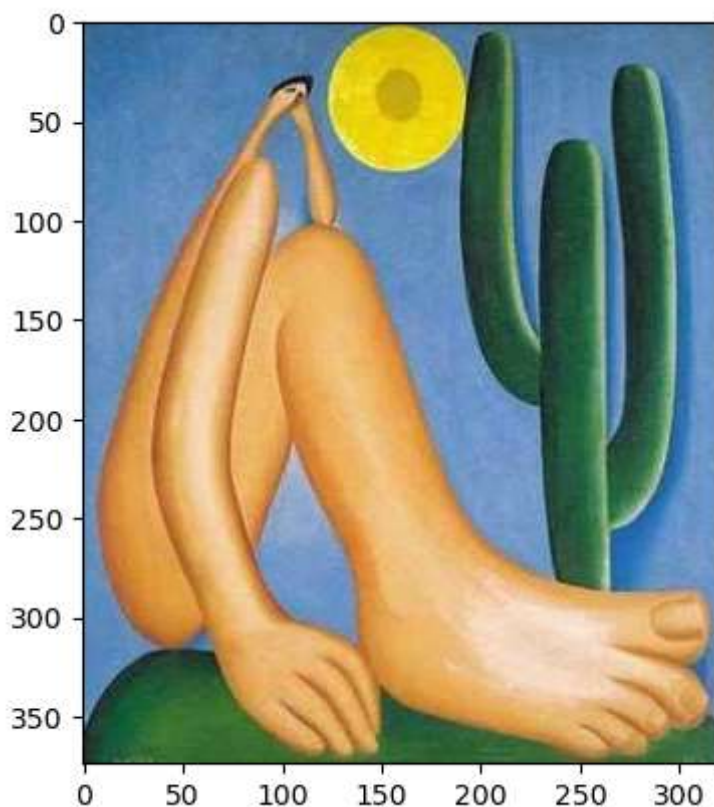
```
Out[4]: <matplotlib.image.AxesImage at 0x2255cbc3b10>
```



**Para visualização, pode-se trocar o índice das matrizes ou apenas usar a função pronta**

```
In [5]: img_rgb = cv2.cvtColor(exemplo, cv2.COLOR_BGR2RGB)  
plt.imshow(img_rgb)
```

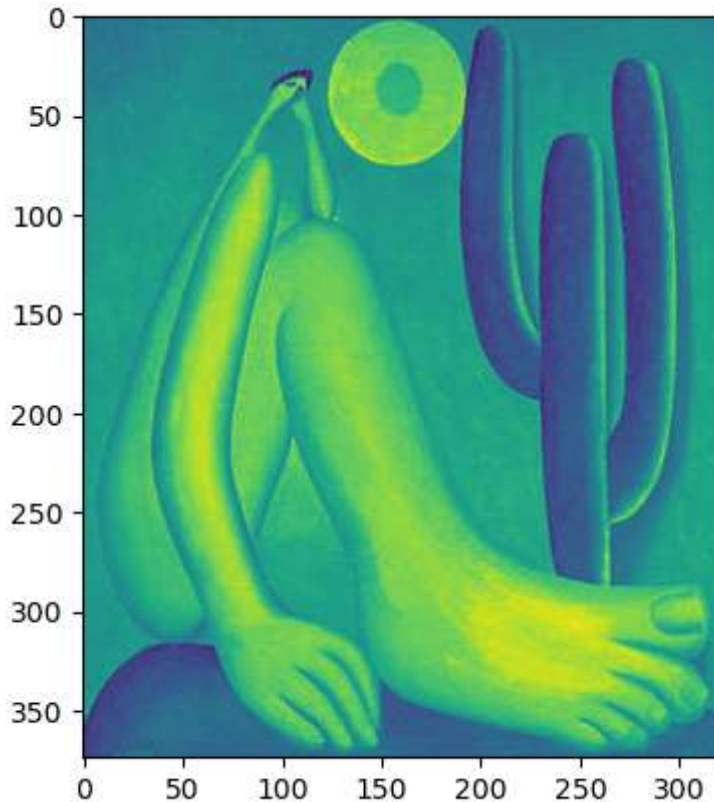
```
Out[5]: <matplotlib.image.AxesImage at 0x2255cc53b10>
```



## Para converter uma imagem em escala de cinza

```
In [6]: img_cinza = cv2.imread('static/exemplo1.jpg',cv2.IMREAD_GRAYSCALE)  
plt.imshow(img_cinza)
```

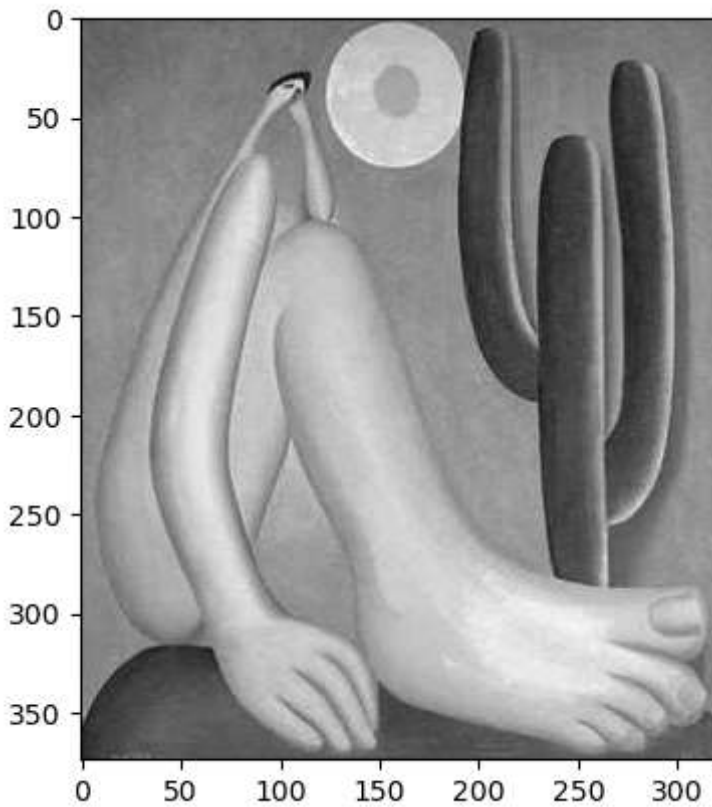
```
Out[6]: <matplotlib.image.AxesImage at 0x2255dd03b10>
```



## Para visualizar a imagem em tons de cinza

```
In [7]: plt.imshow(img_cinza,cmap='gray')
```

```
Out[7]: <matplotlib.image.AxesImage at 0x2255dd8bb10>
```



## Redimensionar a imagem com valores específicos

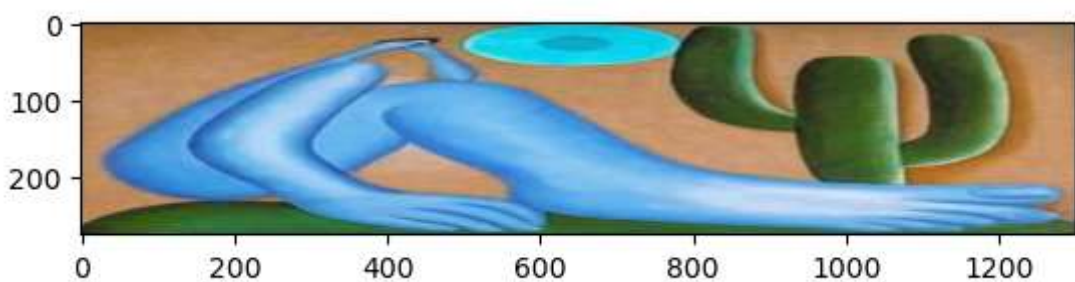
```
In [8]: exemplo.shape
# colunas = largura, linhas = altura, quantidade de matrizes = quantidade de canais
```

```
Out[8]: (374, 320, 3)
```

```
In [9]: achatado = cv2.resize(exemplo,(1300,275))
```

```
In [10]: plt.imshow(achatado)
```

```
Out[10]: <matplotlib.image.AxesImage at 0x2255e614110>
```



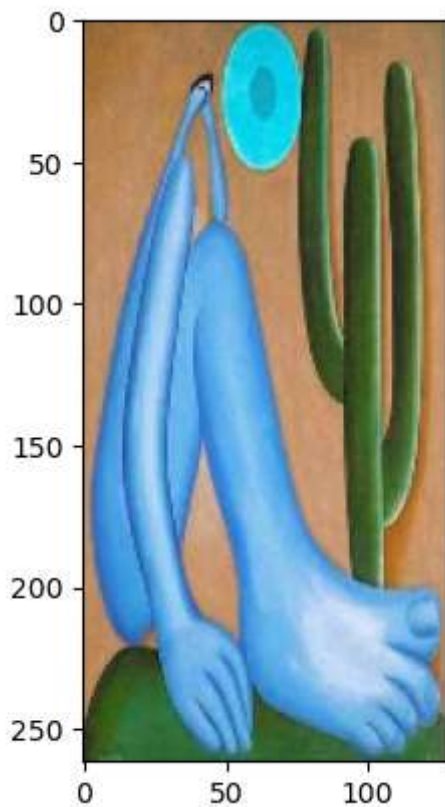
## Redimensionar a imagem por índice (%)

```
In [12]: largura_indice = 0.4 # reduz 60%
         altura_indice = 0.7 # reduz 30%
```

```
In [13]: imagem_reduzida = cv2.resize(exemplo,(0,0),exemplo,largura_indice,altura_indice)
```

```
In [14]: plt.imshow(imagem_reduzida)
```

Out[14]: <matplotlib.image.AxesImage at 0x2255ec94110>



## Rotacionando as imagens

```
In [15]: # em função do eixo x  
rotacao_x = cv2.flip(exemplo,0)  
plt.imshow(rotacao_x)
```

Out[15]: <matplotlib.image.AxesImage at 0x2255f4b3b10>





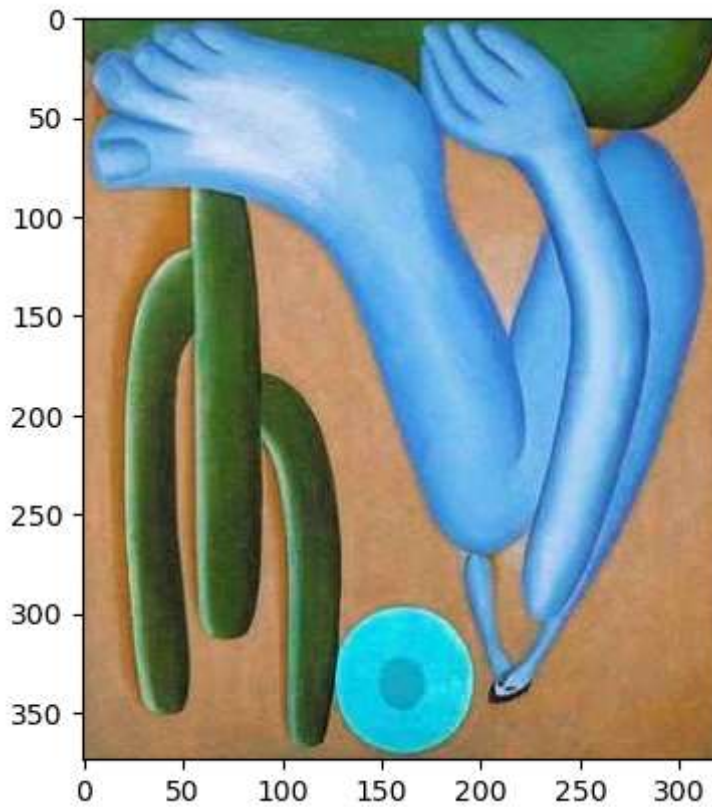
```
In [16]: # em função do eixo y  
rotacao_y = cv2.flip(exemplo,1)  
plt.imshow(rotacao_y)
```

Out[16]: <matplotlib.image.AxesImage at 0x2255f65f790>



```
In [17]: # Na transversal  
rotacao_t = cv2.flip(exemplo,-1)  
plt.imshow(rotacao_t)
```

Out[17]: <matplotlib.image.AxesImage at 0x2255f67d7d0>



## Gravar imagens com o Open CV

```
In [18]: type(rotacao_x)
```

```
Out[18]: numpy.ndarray
```

```
In [48]: cv2.imwrite('salvar_imagem.jpg', rotacao_x)
```

```
Out[48]: True
```

## !!! Cuidado para nao salvar a imagem BGR!!!

## Visualizar a imagem fora do notebook

```
In [39]: %matplotlib qt  
plt.imshow(imagem_reduzida)
```

```
Out[39]: <matplotlib.image.AxesImage at 0x1c27dcc7350>
```

```
In [42]: %matplotlib inline
```