



# Curso de Python

30-01-2021

<https://github.com/tsbressan/CursoPython>

# Bibliotecas para processamento

- **SciPy**: <https://www.scipy.org/>
- Biblioteca completa para matemática, ciência e engenharia.
- Utiliza como base a biblioteca *Numpy* com suporte para trabalhar com grande quantidade de dados
- Necessita ser importada antes da utilização:
- `import scipy as sc`
- Suporte a estatísticas, processamento de sinais e imagens, solução de equações, funções especiais, polinômios,...

# Bibliotecas para processamento

- **SciPy**: <https://www.scipy.org/>
- Principais funções:
- `scipy.stats`, `scipy.interpolate`, `scipy.linalg`, entre outras.

<https://docs.scipy.org/doc/scipy/reference/index.html>

# Bibliotecas para processamento

- **SciPy**: stats
- Pearsonr

```
import scipy.stats as sc
```

```
a = np.array([0, 0, 0, 6, 1, 1, 6])
```

```
b = np.array([0, 8, 0, 8, 1, 1, 4])
```

```
sc.pearsonr(a, b)[0]
```

\*Pode ser utilizado em dados importados (pandas.dataframes)

<https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html#scipy.stats.pearsonr>

# Bibliotecas para processamento

- **SciPy**: stats
- Spearmanr

```
import scipy.stats as sc
```

```
a = np.array([0, 0, 0, 6, 1, 1, 6])
```

```
b = np.array([0, 8, 0, 8, 1, 1, 4])
```

```
sc.spearmanr(a, b)
```

\*Pode ser utilizado em dados importados (pandas.dataframes)

<https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html#scipy.stats.pearsonr>

# Bibliotecas para processamento

- **SciPy:** interpolate
- interp1d

```
from scipy.interpolate import interp1d
import numpy as np
x = np.array([10.2, 20.1, np.nan, 40.4, 65.2, np.nan, 31.1])
```

```
not_nan = np.logical_not(np.isnan(x))
indice = np.arange(len(x))
interp = interp1d(indice[not_nan], x[not_nan])
interp(indice)
```

\*Pode ser utilizado em dados importados (pandas.dataframes)

# Bibliotecas para processamento

- **pandas:** interpolation
- `<dataframe>.interpolate(method='...')`

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.interpolate.html>

# Bibliotecas para processamento

- **SciPy:** ndimage
  - Analise e processamento de imagens 2D (scipy.ndimage)
  - Extrair características, classificar, interpolar, aplicar filtros.
- OpenCV: cv2
  - Biblioteca para processamento gráfico – visão computacional
  - Editar imagem, criar imagem, converter,...



# Bibliotecas para processamento

- **Exemplos Scipy ndimage e cv2**

```
import cv2
```

```
image=cv2.imread('362-u1480e-1h-1-  
a_shlf7852891_20160813075212_trim.jpg')
```

```
type(image)
```

```
print (image)
```

```
Image.shape
```

# Bibliotecas para processamento

- **Exemplos Scipy ndimage e cv2**

# mostrar os canais RGB

red=image[:, :, 0]

green=image[:, :, 1]

blue=image[:, :, 2]

#imagem – escala de cinza

grey = (0.2126 \* red) + (0.7152 \* green) + (0.0722 \* blue)

# Bibliotecas para processamento

- **Exemplos Scipy ndimage e cv2**

```
# mostrar imagem
import matplotlib.pyplot
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
matplotlib.pyplot.imshow(image)
matplotlib.pyplot.show()
cv2.imwrite('original.jpg', cv2.cvtColor(image, cv2.COLOR_BGR2RGB))

import matplotlib.cm as cm
matplotlib.pyplot.imshow(grey, cmap = matplotlib.cm.Greys_r)
matplotlib.pyplot.show()

#Exportar
cv2.imwrite('grey_test.jpg', grey)
```

# Bibliotecas para processamento

- **Exemplos Scipy ndimage e cv2**

## **# aplicar filtro**

```
from scipy.ndimage import gaussian_filter  
import cv2
```

```
image=cv2.imread('362-u1480e-1h-1-  
a_shlf7852891_20160813075212_trim.jpg')  
image = gaussian_filter(image,sigma=5)  
cv2.imwrite('filter_test.jpg', cv2.cvtColor(image,  
cv2.COLOR_BGR2RGB))
```