

	GBC213 – Multimídia	
Atividade 1	CoLab - Introdução	
Prof. Dr. Marcelo Zanchetta do Nascimento		

Informações:

- Deve ser elaborado um arquivo com as soluções no editor do CoLab (Google – arquivo extensão .ipynb). Deve ser colocado comentários nos programas desenvolvidos (use o símbolo #).
- Esse laboratório traz exercícios de revisão a programação Python com bibliotecas de manipulação de mídias.

Exercícios

1) Digite os *arrays* diretamente nos seguintes formatos e verifique os resultados. Para visualizar o resultado use o comando “print”. Para a matriz “C” deve ser importado a biblioteca “**numpy**” com o seguinte comando: “**import numpy as np**”.

- a) `A = [[1, 4, 5, 12], [-5, 8, 9, 0], [-6, 7, 11, 19]]`
- b) `B = [[16, 8, 2, 4],
[5, 7, 8, 11],
[20, 30, 40, 50]]`
- c) `C = np.array([[1, 2, 3], [3, 4, 5]])`
- d) `D = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 8, 8, 9], [8, 7, 7, 8], [4, 5, 9, 8]]`

2) Verifique os resultados de cada linha com as matrizes construídas com objetivo de explorar a indexação.

a) Para selecionar um determinado elemento da matriz deve-se utilizar:

`A[i, j]` sendo *i* número da linha e *j* o número da coluna com *i, j* = 0, ...*n*-1.

Exemplo: Elemento da linha 0 e coluna 1:

```
Al = (A[0][1])
print(Al)
```

b) Usar os dois pontos (:) para representar “todos” ou “até”.

Exemplos:

Todos os elementos da coluna 0:

```
x2 = np.random.randint(10, size=(3, 4))
```

```
print(x2[:, 0])
```

Todos os elementos da primeira linha 0:

```
print(x2[0,:])
```

Todos os elementos da linha 0 até a linha 2 e da coluna 2 até a coluna 3:

```
print(x2[0:2, 2:3])
```

Fazer todos os elementos da coluna 0 iguais a zero:

```
A4 = x2
```

```
A4[:,0] = 0
```

O elemento da última linha e última coluna:

```
A5 = x2[-1,-1]
```

Todos os elementos da última coluna:

```
A6 = x2[:, -1]
```

O elemento da última linha e primeira coluna antes da última:

```
A7 = x2[-1, -2]
```

Matriz transposta de x2:

```
A9 = np.transpose(x2)
```

c) Criar um vetor contendo:

- i) Todos os valores de 1 a 20;
- ii) Todos os valores pares de 1 a 20;
- iii) Todos os valores ímpares de 1 a 20;
- iv) Valores de 0 a 1, intervalados de 0.1.

3) Operações com as matrizes: soma, valor máximo e mínimo.

a) Pode-se obter a soma de todos os elementos da matriz usando:

```
M1 = np.sum(x2)
```

b) Vetor mostrando os valores máximos de cada coluna de uma matriz:

```
M1 = x2.max(axis=0)
```

c) Valor máximo de uma matriz:

```
M2 = np.max(x2)
```

d) Vetor mostrando os valores mínimos de cada coluna de uma matriz:

```
M3 = x2.min(axis=0)
```

e) Valor mínimo de uma matriz:

```
M4 = np.min(x2)
```

f) Dada a matriz E = [1, 2, 3, 4;

5, 6, 7, 8;

9, 10, 11, 12;

13, 14, 15, 16]

- i) Criar a matriz F como uma cópia de E, sem a última linha e coluna;
- ii) Somar "+1" em todos os elementos pares da matriz E;
- iii) Remova a primeira coluna de E.

4) Converta as matrizes para imagens de intensidades e comente os resultados.

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from random import randint
from random import random
from random import seed
from google.colab import drive
drive.mount('/content/gdrive/')

array = (np.random.rand(10, 20)*8).astype(np.uint8)
print(array)

plt.imshow(array, 'gray')
plt.show()

img = Image.fromarray(array)
img.save('sample_data/test1.png')
```

a) Mostre e salve a matriz abaixo em escala de cinza e observe os resultados.

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40

5) No CoLab você pode gerar arranjos padrões que são úteis em diversas aplicações. Use os exemplos abaixo para demonstrar os resultados:

- b) zeros(M,N) à matriz de zeros;
- c) ones(M,N) à matriz de uns;
- d) full(M,N) à matriz de True e False;

e) `random()` à matriz de números aleatórios inteiro em intervalos definidos ou uniformemente distribuídos no intervalo `[0,1]`;

f) `randn(N)` cria uma matriz de forma especificada e a preenche com valores aleatórios conforme a distribuição normal padrão;

g) Explique cada um desses comandos e comente os resultados.

I. `Z1 = np.zeros((5,), dtype=int)`

II. `Z2= np.ones(5)`

III. `value = randint(2, 5)`

IV. `seed(1)`

V. `print(random())`

VI. `np.random.randn(5)`

6) Leia e mostre as imagens verificando as informações sobre elas. (Obs: Se o arquivo não estiver na pasta `sample_data` é preciso informar o caminho, ou salvar a imagem no diretório). Neste exemplo use as imagens `rose_gray.tif` e `test1.png` que estão disponíveis no Moodle-UFU.

```
Import numpy as np
import matplotlib.pyplot as plt
import imageio as im

from google.colab import drive
drive.mount('/content/gdrive/')

image = im.imread('sample_data/test1.png')
plt.imshow(image, cmap='gray')
image.size
```

```
From matplotlib import pyplot as plt
import cv2

img = cv2.imread('sample_data/test1.png', 0)
plt.imshow(img)
plt.show()
```

a) Verifique o número de linhas e colunas da imagem:

```
print(im.shape)
```

b) Atribua os valores das linhas e colunas às variáveis `M` e `N`:

```
[M,N] = img.shape
```

```
print (M,N)
```

7) Construa programas que empregue as bibliotecas em Python (scikit-image, SciPy e SimpleCV) para abrir novas imagens armazenadas no CoLab. Sobre as imagens crie um código que remova 50 pixels de cada lado da imagem e mostre o resultado.

8) No caso de um arquivo de áudio é possível observar as informações:

```
import wave

arquivoWav = wave.open('sample_data/Vocal.wav', 'r')
print('Número canais:', arquivoWav.getnchannels())
print('Compactação:', arquivoWav.getcompname())
```

Crie um programa que leia um arquivo e mostre as informações sobre o número de canais e taxa de amostragem?

9) Plote os gráficos das seguintes equações

- a) $y = \sin(x) + 2$, para o intervalo $[-2\pi, 2\pi]$
- b) $y = 2x + 3$, para o intervalo $[0, 100]$
- c) $y = x^2 + 4x - 17$, para o intervalo $[-100, 100]$