Lógica de Programação Orientada a Objetos

Bem-vindos!

Objetivos

- ✔ Numpy

LET'S CODE

O que é o NumPy

- O NumPy é uma poderosa biblioteca Python que é usada principalmente para realizar cálculos em Arrays Multidimensionais. O NumPy fornece um grande conjunto de funções e operações de biblioteca que ajudam os programadores a executar facilmente cálculos numéricos.

Instalação da biblioteca NumPy

- pip install numpy

Arrays NumPy Unidimensional

```
import numpy as np
my_array = np.array([1, 2, 3, 4, 5])
print(my_array)

print(my_array.shape)
```

Arrays NumPy Multidimensionais

- ```
  my_array = np.array([[1, 2, 3, 4, 5], [1, 2, 3, 4, 5]])
  print(my_array[0])
  print(my_array[1])
  print(my_array.shape)
  my_array[0] = -1
  print(my_array)
  ```

- ```
  a = np.arange(6).reshape((3, 2))
  print(a)
  ```

- ```
  my_array = np.array([[4, 5], [6, 1]])
  print(my_array[0][1])
  ```

- ```
  my_array = np.array([[4, 5], [6, 1]])
  print(my_array)
  ```

Iniciando um array com valores aleatórios

- ```
  import numpy as np
  x = np.empty([3,2], dtype = int)
  print(x)
  ```

- ```
  my_random_array = np.random.random((5))
  print(my_random_array)
  ```

- ```
  a = np.floor(10*np.random.random((3,4)))
  print(a)
  ```
  ```

Iniciando um array de 0 e de 1

- ```
  my_new_array = np.zeros((5))
  print(my_new_array)
  ```

- ```
  my_new_array = np.ones((5))
  print(my_new_array)
  ```

Indexação e slicing

- my_array2 = np.array([3, 2, 8, 22, 127])
  my_array2[3]
- my_array2 = np.array([3, 2, 8, 22, 127])
  print(my_array2[1:])
- my_array2 = np.array([3, 2, 8, 22, 127])
  print(my_array2[2:4])

Manipulando array em NumPy

```
import numpy as np

a = np.array([[1.0, 2.0], [3.0, 4.0]])
b = np.array([[5.0, 6.0], [7.0, 8.0]])

sum = a + b # Soma
difference = a - b # Subtração
product = a * b # Multiplicação
quotient = a / b # Divisão

print('Sum = \n', + sum)
print('\n')
print('Difference = \n', + difference)
print('\n')
print('Product = \n', + product)
print('\n')
print('Quotient = \n', + quotient)
```

Multiplicação de arrays

```
import numpy as np

a = np.array([[1.0, 2.0], [3.0, 4.0]])
b = np.array([[5.0, 6.0], [7.0, 8.0]])
matrix_product = a.dot(b)

print('Matrix Product = \n', + matrix_product)
```

EXERCÍCIO 1

Remova todos os números ímpares de um NumPy Array que vai de 0 a 20

EXERCÍCIO 2

Dado um NumPy Array unidimensional que vai de 0 a 8, converta-o em um NumPy Array bidimensional e quadrado.

EXERCÍCIO 3

Dado 2 NumPy Array, sendo o primeiro de tamanho 2 linhas por 5 colunas, com valores de 0 a 9 e outro de tamanho 3 linhas por 5 colunas, com valores de 10 a 24, crie um terceiro NumPy Array de tamanho 5 linhas por 5 colunas que concatene verticalmente os NumPy Arrays 1 e 2.
Dica: Pesquise as funções hstack e vstack.

EXERCÍCIO 4

Dados 2 NumPy Arrays:
Array1: [0, 10, 20, 40, 60]
Array2: [10, 30, 40]
Faça uma função que retorne os valores iguais entre os 2 arrays.
Dica: Use a função intersect1d.

EXERCÍCIO 5

Dada a matriz:

```
array = np.array([[1,2,3,4],
                  [1,2,3,4],
                  [5,6,7,8],
                  [1,2,3,4],
                  [3,3,3,3],
                  [5,6,7,8]])
```

Faça uma função que remova as linhas duplicadas, deixando apenas uma incidência da mesma.
Dica: Use a função unique.