



# ASTROINFORMÁTICA I

## AULA 03

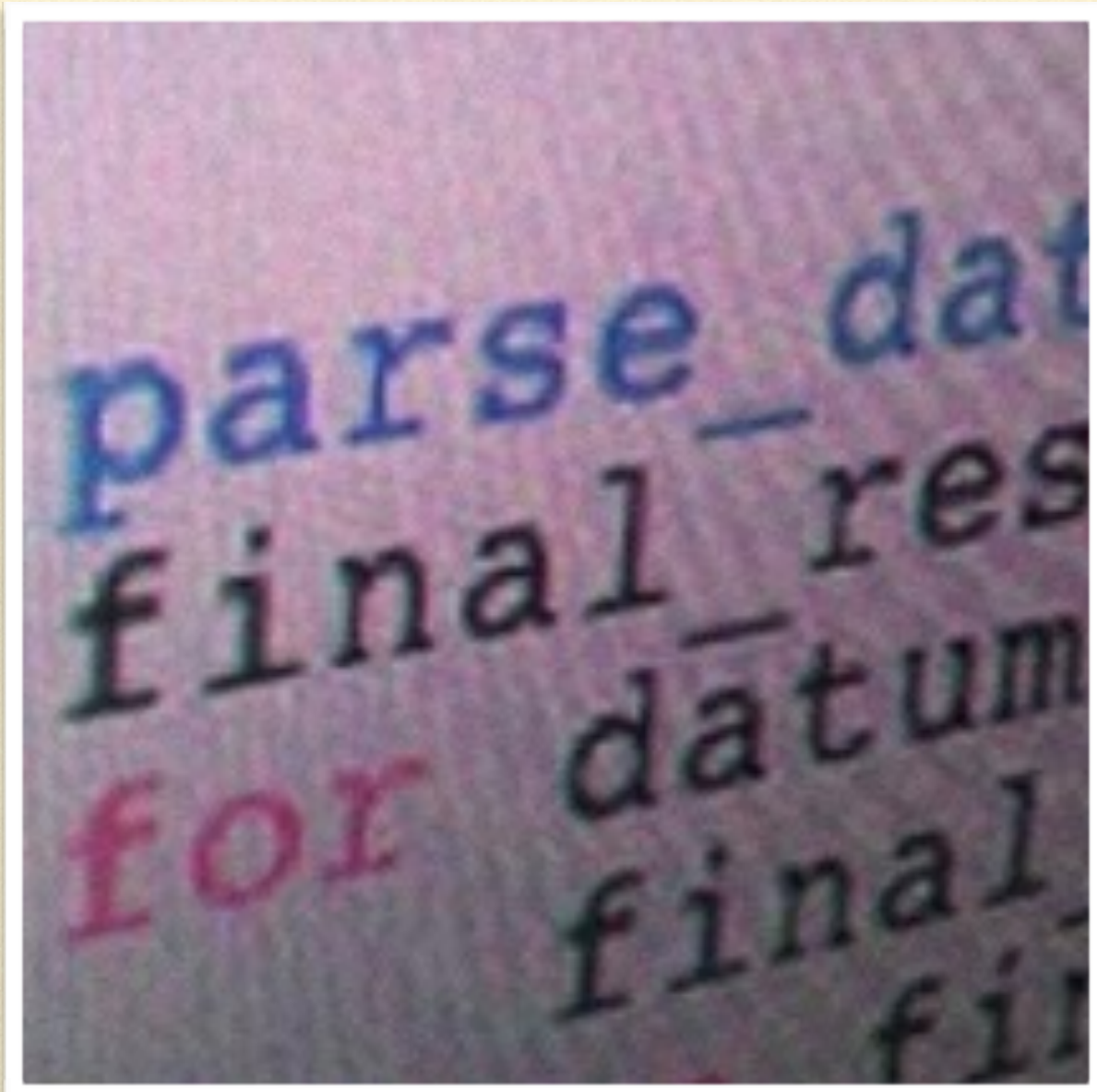
Prof. Dr. Luciano Silva

[luciano.silva@mackenzie.br](mailto:luciano.silva@mackenzie.br)



# OBJETIVOS

---



- Conhecer os conceitos de vetores e matrizes em Python
- Aprender a manipular vetores e matrizes em Python em baixo nível
- Aprender a manipular vetores e matrizes utilizando o módulo NUMPY



---

# VETORES (ARRAYS)

## *LISTAS LINEARES INDEXADAS*

---

Num vetor (ou lista linear indexadas), os elementos são acessados através de índices

```
>>> languages = ["Python", "C", "C++", "Java", "Perl"]
>>> print(languages[0] + " and " + languages[1] + " are quite different!")
Python and C are quite different!
>>> print("Accessing the last element of the list: " + languages[-1])
Accessing the last element of the list: Perl
>>>
```

Vetores podem ser heterogêneos (elementos de tipos diferentes)

```
group = ["Bob", 23, "George", 72, "Myriam", 29]
```

---



---

# EXERCÍCIO I

---

Definimos o produto escalar entre dois vetores pela soma abaixo:

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^n A_i B_i = A_1 B_1 + A_2 B_2 + \cdots + A_n B_n$$

Construa um programa em Python que calcule o produto escalar entre os vetores A e B mostrados abaixo:

$$\mathbf{A}=[1,2,3,4,5] \text{ e } \mathbf{B}=[6,7,8,9,10]$$

---



# VETORES EM NUMPY

O módulo NUMPY possui rotinas básicas para manipulação de vetores e matrizes em Python

```
import numpy as np
x = np.array([42,47,11], int)
x
array([42, 47, 11])
```

```
>>> x = np.array([1,5,2])
>>> y = np.array([7,4,1])
>>> x + y
array([8, 9, 3])
>>> x * y
array([ 7, 20,  2])
>>> x - y
array([-6,  1,  1])
>>> x / y
array([0, 1, 2])
>>> x % y
array([1, 1, 0])
```



---

# EXERCÍCIO II

---

Definimos o produto escalar entre dois vetores pela soma abaixo:

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^n A_i B_i = A_1 B_1 + A_2 B_2 + \cdots + A_n B_n$$

Construa um programa em Python, utilizando o módulo **NUMPY**, que calcule o produto escalar entre os vetores A e B mostrados abaixo:

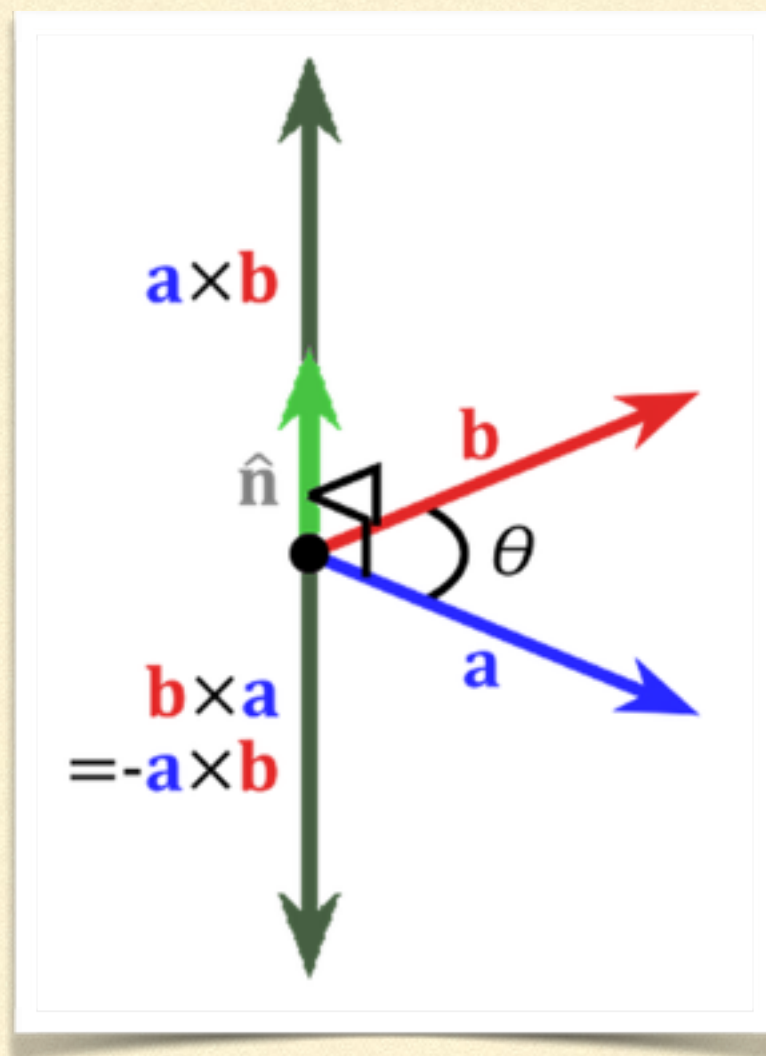
$$A=[1,2,3,4,5] \text{ e } B=[6,7,8,9,10]$$

---



# EXERCÍCIO III

O produto vetorial entre dois vetores  $a$  e  $b$ , denotado por  $a \times b$ , é um vetor perpendicular ao plano definido por  $a$  e  $b$ .



Construa um programa em Python, utilizando o módulo **NUMPY**, que calcule o produto vetorial entre os vetores (versores)  $a$  e  $b$  abaixo:

$$A=[1,0,0] \text{ e } B=[0,1,0]$$



---

# MATRIZES (ARRAYS BIDIMENSIONAIS)

## LISTAS DE LISTAS

---

Uma matriz em Python pode ser vista como uma lista de listas:

```
>>> matrix = [  
...     [1, 2, 3, 4],  
...     [5, 6, 7, 8],  
...     [9, 10, 11, 12],  
... ]
```

Para acessar um determinado elemento, acessamos a linha (lista) e depois a posição dentro desta lista (coluna).

---



---

# EXERCÍCIO III

---

Calcular a transposta da matriz abaixo:

```
>>> matrix = [  
...     [1, 2, 3, 4],  
...     [5, 6, 7, 8],  
...     [9, 10, 11, 12],  
...     ]
```



# EXERCÍCIO III (RESPOSTA)

```
>>> transposed = []
>>> for i in range(4):
...     transposed.append([row[i] for row in matrix])
...
>>> transposed
[[1, 5, 9], [2, 6, 10], [3, 7, 11], [4, 8, 12]]
```



---

# EXERCÍCIO IV

---

Calcular a transposta da matriz abaixo em NUMPY:

```
>>> matrix=np.array(((1,2,3,4),(5,6,7,8),(9,10,11,12)))
>>> matrix
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```



# MULTIPLICAÇÃO MATRIZ-VETOR

```
>>> matrix
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
>>> vector
array([1, 1, 1, 1])
>>> np.dot(matrix,vector)
array([10, 26, 42])
```