**Questões Objetivas – Aula 09**

Objetivos de aprendizagem:

* Listar as aplicações da biblioteca.
* Listar as funções da biblioteca.
* Definir as estruturas necessárias para utilizar a biblioteca.
* Utilizar a biblioteca para aplicações relacionadas a neuroengenharia.

**Questão 01** -Assinale a alternativa INCORRETA sobre a biblioteca NumPy.

1. O objetivo principal do NumPy é o vetor multidimensional.
2. **As operações realizadas pelo NumPy não podem ser salvas em arquivos de formato csv e txt.**
3. O NumPy realiza operações com vetores e matrizes, onde estas operações são realizadas ponto a ponto ou não.
4. Para utilizar o NumPy basta instalar a biblioteca e utilizar o comando: import numpy as np

**Questão 02 –** Na figura abaixo a saída esperada é:

import numpy as np

vetor = np.array(((1., 0., 0.),(0.,1.,2.)))

print(vetor)

1. **[[1. 0. 0.]**

**[0. 1. 2.]]**

b) [[1.+0.j 0.+0.j 0.+0.j]

[0.+0.j 1.+0.j 2.+0.j]]

c) [[0.+0.j 1.+0.j 2.+0.j]

[1.+0.j 0.+0.j 0.+0.j]]

d) [[0. 1. 2.]

[1. 0. 0.]]

**Questão 03 –** Na imagem abaixo foi gerada duas matrizes com o comando np.array. Qual será o comando para criar uma matriz transposta de B e imprimir?

import numpy as np

A = np.array([[1,1],

             [0,1]])

B = np.array([[2,0],

              [3,4]])

1. Print(A\*B)
2. Print(A@B)
3. **Print(np.transpose(B))**
4. Print(@B)

**Questão 04 –** Um aluno do mestrado em neuroengenharia do ISD está realizando uma pesquisa através da captura do movimento e foi gerado alguns ângulos desse movimento. Sabendo disso, ele gostaria de gerar os dados com o maior e o menor ângulo obtido com 2 casas decimais. Qual seria a função da biblioteca numpy que ele utilizaria?

1. **Maior ângulo: np.around(ângulo.max(),decimals=2)) e Menor ângulo: np.around(ângulo.min(),decimals=2))**
2. Maior ângulo: np.around(ângulo.max(),decimals=3)) e Menor ângulo: np.around(ângulo.min(),decimals=3))
3. Maior ângulo: np.around(ângulo.max(),decimals=2)) e Menor ângulo: np.around(ângulo.mean(),decimals=2))
4. Maior ângulo: np.around(ângulo.sum(),decimals=2)) e Menor ângulo: np.around(ângulo.trapz(),decimals=2))