

**LIBRERIAS**

import numpy as np

import matplotlib.pyplot as plt

**DIMENSIONES Y ELEMENTOS**

vector 1 grado de libertad , x = conj de escalares (ARRAY)

matriz 2 grados de libertad x,y

tensor 1 mas grados de libertad x,y,z

len(vector) = no. elementos

matriz.size = No. elementos

matriz.shape = dimensiones x - y

tensor.shape = dimensiones x - y – z

vector = np.array([1,2,3,4])

4x3

matriz = np.array([1,2,3,4],[1,2,3,4], [1,2,3,4])

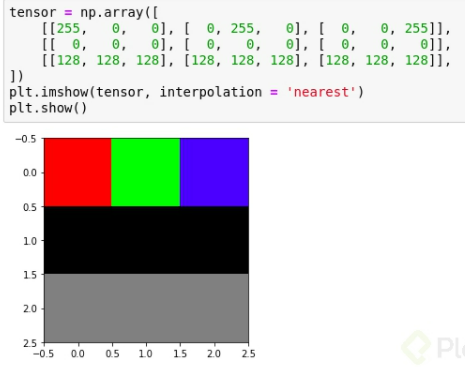
4x3x2

tensor = np.array([

[ [1,2,3,4], [1,2,3,4], [1,2,3,4] ],

[ [1,2,3,4], [1,2,3,4], [1,2,3,4] ]

])



Black (0,0,0)

White (255,255,255)

Red (255,0,0)

Green (0,255,0)

Blue (0,0,255)

#tensor[0,:,:] recupera la matriz 1

# tensor[1,:,:] recupera la matriz 2

# tensor[2,:,:] recupera la tercera matriz

**GENERAR ALEATORIAMENTE**

generarlos aleatoriamente.

v1 = np.random.randint(min\_val, max\_val, size=(x,y,z))

min\_val = valor minimo que comprende la funcion  
max\_val = valor maximo que comprende la funcion

x = numero de filas  
y = numero de columnas  
z = numero de dimensiones (estas seran iguales y tiene como predeterminado el valor de 1)

**TRASPONER, CAMBIAR FILAS POR COLUMNAS**

|  |  |
| --- | --- |
| Mat = np.array( [  [[5,4,3], [2,1,0]]  ]) | Mat.T |
| 5 4 3  2 1 0 | 5, 2  4, 1  3, 0 |

La traspuesta de A.dot(B) = A.dot(B).T = B.T.dot(A.T)

**MATRIZ IDENTIDAD**

Identidad = np.eye(4)

1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1

Identidad.dot(Vector) = Vector

A \* A-1 = identidad

**MATRIZ QUE NO TIENE INVERSA PORQUE UNA FILA ES DEPENDIENTE DE OTRA**

import numpy as np

A= np.array([

[ 0, 1, 0, 0]

[ 0, 0, 1, 0]

[ 0, 1, 1, 0]

[ 1, 0, 0, 1]

])

lamdas, V= np.linalg.eig(A.T)

print (A[lambdas ==0, :])

R/

[ 0, 1, 1, 0]

La fila 3 es dependiente, es la suma de la 1 y de la 2 => no tiene inversa

Mariz singular, porque no tiene inversa, tiene filas dependientes

**np.ling.inv(A) nos da error**