Ejercicio Numpy-01

Crea un array 3x4 con valores enteros aleatorios entre 10 y 50. Luego, calcula la suma, la media, y el valor máximo de todo el array.

Ejercicio Numpy-02

Crea una matriz identidad 5x5 y cambia la diagonal principal a un valor de 7.

Ejercicio Numpy-03

Dado el array *a = np.arange(20)*, crea un nuevo array con los valores pares mayores que 5 usando boolean masking.

Ejercicio Numpy-04

Dado el array bidimensional siguiente:

a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

Reemplaza todas las entradas de la segunda fila por 0 y luego aplana el array a una dimensión.

Ejercicio Numpy-05

Crea dos arrays 1D a y b con valores [1, 2, 3] y [4, 5, 6]. Realiza el producto cruzado, producto escalar y producto externo.

Ejercicio Numpy-06

Crea un array de 10 elementos que vayan de 0 a 100 de forma equiespaciada, redondea cada elemento a su entero más cercano, y luego limita sus valores para que estén entre 20 y 80.

Ejercicio Numpy-07

Crea una matriz aleatoria 4x4 con valores entre 0 y 1. Multiplica la matriz por sí misma (producto matricial) y calcula la traza del resultado.

Ejercicio Numpy-08

Dado el array *a = np.array([10, 20, 30, 40, 50])*, crea una máscara booleana para seleccionar los elementos que no estén en la lista *[20, 40]*.

Ejercicio Numpy-09

Crea un array 2x3 lleno de 5s, otro 3x2 lleno de 10s, y concaténalos verticalmente tras hacer el reshape adecuado para que la concatenación sea posible.

Ejercicio Numpy-10

Genera un array de 3x3 con valores aleatorios enteros entre 1 y 9. Calcula sus autovalores y autovectores, y verifica que multiplicar la matriz por un autovector da el autovalor multiplicado por el autovector.

Ejercicio Numpy-11

Crea un sistema de ecuaciones 3x3 y resuélvelo usando np.linalg.solve. El sistema es:

$$2x + 3y - z = 5$$
$$x + 7y + 2z = 3$$
$$4x - 5y + 6z = 10$$

Ejercicio Numpy-12

Genera un array de 50 números aleatorios del 0 al 100. Separa el array en 5 partes iguales y calcula la media de cada una.

Ejercicio Numpy-13

Dado un array 2D *a = np.arange(1, 17).reshape(4,4)*, intercambia filas y columnas (transpuesta), luego ordena cada fila y devuelve los índices de ordenación

Ejercicio Numpy-14

Crea un array 1D con valores del 1 al 15. Itera sobre él con np.nditer y crea otro array que contenga el cuadrado de cada elemento.

Ejercicio Numpy-15

Crea una matriz 4x4 con valores aleatorios enteros entre 0 y 9. Usa np.where para cambiar todos los valores menores que 5 por -1, y los demás déjalos iguales

Ejercicio Numpy-16

Eres un ingeniero que debe resolver un sistema lineal de 4 ecuaciones con 4 incógnitas para modelar el equilibrio de fuerzas en una estructura. El sistema es:

$$3x+y-z+2w=52$$
$$x-2y+4z-w=-2$$
$$-x+21y-z+w=0$$
$$x+y+z+w=3$$

Además, calcula la inversa de la matriz de coeficientes y comprueba que la multiplicación entre matriz e inversa da la matriz identidad.

Ejercicio Numpy-17

Como científico de datos, generas un conjunto de vectores 3D aleatorios para analizar su orientación. Genera 10 vectores 3D con valores aleatorios entre -10 y 10. Para cada vector, calcula su módulo (norma) y normalízalo (vector unitario).

Después, calcula la matriz de producto externo para cada vector normalizado y almacena todas esas matrices en un array 3D de dimensiones (10, 3, 3).

Ejercicio Numpy-18

Un físico está estudiando un sistema cuántico con una matriz Hamiltoniana hermítica 4x4 generada aleatoriamente. Genera esta matriz hermítica (simétrica y con valores complejos conjugados) con números complejos aleatorios. Calcula los autovalores y autovectores, y verifica que la matriz es diagonalizable, es decir, que H @ V = V @ D, donde D es la matriz diagonal de autovalores.

Ejercicio Numpy-19

Un investigador analiza la evolución temporal de un sistema descrito por una matriz de transición 5x5. Genera una matriz estocástica (cada fila suma 1 y todos los valores son positivos) y calcula su estado estable como el autovector asociado al autovalor 1, normalizado para que la suma de sus componentes sea 1.

Ejercicio Numpy-20

Un matemático quiere comprobar la propiedad de ortogonalidad de un conjunto de vectores generados mediante la función np.eye con pequeñas perturbaciones gaussianas añadidas. Crea una matriz 6x6 identidad y suma a cada elemento un ruido normal con media 0 y desviación 0.01. Calcula la matriz producto $M=A^TA$. Comprueba qué tan cerca está M de la matriz identidad calculando la norma de la diferencia y muestra el resultado.