

Working Time 😎

L1: La librería numpy

Es hora de que pongas en práctica todo lo aprendido. 😊

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes **desafíos** que podrás resolver de forma independiente y a tu ritmo.

Más adelante conseguirás las soluciones para que valides tus respuestas y puedas monitorear tu progreso. 😊

¡Manos a la obra!

1. Desafío ⏱

Este desafío consiste en la creación y manipulación de **arrays y matrices** en **NumPy**, así como la aplicación de operaciones matemáticas y algebraicas avanzadas.

- **Creación y manipulación de arrays:**
 - Genera un array unidimensional de **20 números enteros aleatorios** entre 1 y 100.
 - Calcula la **suma, promedio, máximo y mínimo** del array.
 - Ordena el array en **forma ascendente y descendente**.
 - Extrae todos los valores **pares** del array.
 - Reemplaza todos los valores **impares** por -1.
- **Operaciones con matrices:**
 - Crea dos matrices (**4x4**) de **números enteros aleatorios** entre 1 y 50.
 - Realiza la **suma, resta y multiplicación matricial** entre ambas matrices.
 - Calcula la **matriz transpuesta** de la primera matriz.
 - Obtén el **determinante** de la segunda matriz.
 - Encuentra la **inversa** de la segunda matriz (si es invertible).
- **Aplicación de funciones en NumPy:**
 - Genera un array de **100 valores distribuidos uniformemente** entre 0 y 10.
 - Calcula el **seno y coseno** de cada valor en el array.
 - Aplica la función **exponencial** a cada elemento del array.
 - Encuentra la **raíz cuadrada** de cada elemento mayor a 5 en el array.

2. ¿Dónde se lleva a cabo?

Visual studio code

3. Tiempo de dedicación

2 horas

4. Recursos

Documentación de NumPy: <https://numpy.org/doc/>

5. Plus

- *Escribe funciones que realicen cada operación de manera modular.*
- *Prueba con diferentes tamaños de matrices y analiza cómo cambian los resultados.*
- *Experimenta aplicando NumPy a conjuntos de datos reales.*

6. Condición

Esta práctica o ejercitación **no requiere ser entregada y/o evaluada** por el mentor. No obstante puedes compartir tus resultados con el resto de los bootcampers y construir conocimiento en conjunto.

Resolución del desafío

1. Creación y manipulación de arrays en NumPy

```
import numpy as np

# Crear un array de 20 números enteros aleatorios entre 1 y 100
array = np.random.randint(1, 101, size=20)
print("Array:", array)

# Obtener estadísticas básicas
print("Suma:", np.sum(array))
print("Promedio:", np.mean(array))
print("Máximo:", np.max(array), "- Mínimo:", np.min(array))

# Ordenar de forma ascendente y descendente
print("Orden Ascendente:", np.sort(array))
print("Orden Descendente:", np.sort(array)[::-1])

# Filtrar valores pares y modificar impares
pares = array[array % 2 == 0]
array[array % 2 != 0] = -1
print("Valores Pares:", pares)
print("Array modificado:", array)
```

2. Operaciones con matrices en NumPy

```

# Crear dos matrices aleatorias 4x4
matriz1 = np.random.randint(1, 51, size=(4, 4))
matriz2 = np.random.randint(1, 51, size=(4, 4))
print("Matriz 1:\n", matriz1)
print("Matriz 2:\n", matriz2)

# Operaciones básicas
print("Suma:\n", matriz1 + matriz2)
print("Resta:\n", matriz1 - matriz2)
print("Multiplicación Matricial:\n", np.dot(matriz1, matriz2))

# Transpuesta, determinante e inversa
print("Transpuesta de Matriz 1:\n", matriz1.T)
print("Determinante de Matriz 2:", np.linalg.det(matriz2))

# Verificar si la matriz 2 es invertible antes de calcular su inversa
if np.linalg.det(matriz2) != 0:
    print("Inversa de Matriz 2:\n", np.linalg.inv(matriz2))
else:
    print("Matriz 2 no es invertible.")

```

3. Aplicación de funciones en NumPy

```

# Generar un array de 100 valores entre 0 y 10
datos = np.linspace(0, 10, 100)

# Aplicar funciones matemáticas
seno = np.sin(datos)
coseno = np.cos(datos)
exp = np.exp(datos)
raiz_cuadrada = np.sqrt(datos[datos > 5])

print("Seno de los valores:", seno)
print("Coseno de los valores:", coseno)
print("Exponencial de los valores:", exp)
print("Raíz cuadrada de valores mayores a 5:", raiz_cuadrada)

```