

Taller número 3. Programación Numérica con Python y NumPy.

Descripción: Este taller está diseñado para introducir y afianzar el uso de la biblioteca NumPy en Python, con el objetivo de desarrollar habilidades en la creación y manipulación de arrays numéricos, realizar operaciones matemáticas básicas y complejas, y gestionar datos con Python para aplicaciones en ciencia de datos, ingeniería y matemáticas.

Objetivos del Taller:

- Familiarizar a los participantes con la creación y manipulación de arrays en NumPy.
- Desarrollar habilidades para realizar operaciones matemáticas básicas con arrays.
- Aplicar técnicas de indexación y slicing para el manejo efectivo de subconjuntos de datos.
- Utilizar broadcasting y funciones universales para simplificar operaciones matemáticas.
- Aprender métodos de manipulación de formas y álgebra lineal en matrices.
- Manejar y tratar datos faltantes dentro de arrays numéricos.
- Practicar el almacenamiento y carga de arrays para su uso en análisis de datos.

Metodología:

- Se combinarán breves sesiones teóricas con prácticas guiadas en computadora.
- Los ejercicios se resolverán paso a paso, con explicaciones detalladas y asistencia individualizada.
- Se promoverá la discusión grupal para explorar diferentes soluciones y enfoques.

Recursos:

- Computadoras con Python y NumPy instalados.
- Acceso a documentación en línea de NumPy y otros recursos educativos.
- Cuadernos de trabajo o espacios de codificación en línea como Jupyter Notebooks.

Evaluación:

- Se revisará la corrección de los ejercicios individuales.
- Se valorará la capacidad para integrar los conceptos aprendidos en un programa final con menú interactivo.
- Se alentará a los participantes a explicar su código y el razonamiento detrás de sus soluciones.

Contenido del Taller:

Ejercicio 1: Creación y Propiedades de Arrays

Pistas:

- Usa `np.array(range(1, 11))` para crear el array.
- Para cambiar la forma, emplea el método `.reshape(2, 5)` en el array creado.
- Accede a las propiedades `.shape`, `.size`, y `.ndim` para imprimir las dimensiones, la forma y el tamaño.

Ejercicio 2: Operaciones Básicas

Pistas:

- La suma y la resta se pueden hacer directamente ($a + b$, $a - b$).
- Para el producto elemento a elemento, simplemente multiplica a y b .
- Utiliza `np.sum(a)` para sumar todos los elementos dentro de a .

Ejercicio 3: Indexación y Slicing

Pistas:

- Recuerda que en Python, el índice comienza en 0, así que el quinto elemento es `data[4]`.
- Para obtener una subsección, utiliza `data[2:7]`.

Ejercicio 4: Broadcasting y Funciones Universales

Pistas:

- Crea A usando `np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`.
- Para sumar un escalar a A y aplicar una ufunc, simplemente haz $A + 10$ y `np.sqrt(A)`.

Ejercicio 5: Manipulación de Formas y Álgebra Lineal

Pistas:

- Usa `.reshape(3, 2)` para cambiar la forma de M .
- El producto punto de M y su transpuesta ($M.T$) se puede calcular con `np.dot(M, M.T)`.

Ejercicio 6: Trabajo con Datos Faltantes

Pistas:

- Utiliza `np.nan_to_num(data, nan=0)` para reemplazar `np.nan` por 0.
- Calcula la media del array resultante con `np.mean(data)`.

Ejercicio 7: Guardar y Cargar Arrays

Pistas:

- `np.save('mi_array.npy', data)` guarda el array `data`.
- Usa `np.load('mi_array.npy')` para cargar el array desde el archivo.

Programa Integrador con Menú de Opciones: Desarrollar un programa que incorpore todos los ejercicios anteriores en un menú interactivo, reforzando así las habilidades de programación y consolidando el conocimiento adquirido.