Python

ACTIVIDAD PRÁCTICA

Reto 7

# Título: NUMPY

# Equipo de trabajo:

Presentado por: **Ana Milena Rosero Unigarro**

Presentado a Profesora: **Sara Galván Ortega**

# Preguntas de interés:

* ¿Qué hace la librería Numpy?

La librería NumPy (Numerical Python) es fundamental en el manejo de Python, se utiliza para el procesamiento numérico de datos. NumPy proporciona soporte para trabajar con arreglos multidimensionales (arrays) y funciones matemáticas que operan en estos arreglos.

Algunas de las principales funciones y características de NumPy:

* Arreglos Numéricos: que permite manejar de manera más eficiente las listas de Python, almacecnandp datos numéricos.
* Operaciones Matemáticas: NumPy proporciona un amplio conjunto de funciones matemáticas para realizar operaciones numéricas en arreglos, incluyendo sumas, restas, multiplicaciones, divisiones, exponentes, trigonometría, funciones estadísticas y más.
* Indexación: se puede acceder a elementos específicos en un arreglo NumPy de manera eficiente.
* Broadcasting: NumPy permite realizar operaciones entre arreglos de diferentes tamaños y formas de manera automática y eficiente mediante un mecanismo llamado "broadcasting."
* Funciones de Álgebra Lineal: NumPy incluye funciones para realizar operaciones de álgebra lineal, como multiplicación de matrices, descomposiciones, resolución de sistemas de ecuaciones lineales, cálculo de determinantes y valores propios, entre otros.
* Entrada/Salida de Datos: NumPy proporciona funciones para cargar y guardar datos desde/hacia archivos de texto y binarios.
* ¿En qué situaciones se debe usar la librería Numpy?

NumPy es una librería que se utiliza en una amplia variedad de situaciones, especialmente en la programación, el análisis de datos y la operaciones matemáticas en Python.

Especialmente se puede usar Mumpy para:

* Manipulación de Datos Numéricos: NumPy es ideal para trabajar con datos numéricos, como matrices, arreglos multidimensionales. Se puede realizar operaciones matemáticas, estadísticas y algebraicas en estos datos de manera eficiente.
* Procesamiento de Imágenes y Visión por Computadora: NumPy se utiliza en aplicaciones de procesamiento de imágenes para realizar filtrado, análisis de bordes, segmentación y manipulación de imágenes.
* Simulación y Modelado: NumPy es útil para crear simulaciones numéricas y modelar sistemas complejos en campos como la física, la economía y la ecología.

NumPy es una librería muy poderosa que de acuerdo con la documentación se utiliza en una amplia gama de aplicaciones científicas y técnicas en Python.

* ¿En qué situaciones no debe usarse la librería Numpy?

Pueden existir situaciones en las que podría no ser la elección más adecuada o en las que no sea necesario utilizar NumPy. Entre ellas, se presentan algunas situaciones:

* Proyectos Pequeños o Sencillos: En proyectos pequeños o tareas sencillas que no implican un procesamiento numérico intensivo, cargar la librería NumPy podría ser innecesario y aumentar la complejidad del código.
* Aplicaciones Web Frontend: En aplicaciones web frontend, donde JavaScript es el lenguaje predominante para el procesamiento del lado del cliente, NumPy no es relevante. JavaScript proporciona sus propias herramientas para trabajar con datos en el navegador web.
* Programación Web Backend: En el backend de aplicaciones web, donde Python se utiliza para manejar solicitudes HTTP y lógica de servidor, NumPy no suele ser necesario a menos que estés realizando cálculos numéricos específicos en el servidor.
* Aplicaciones de Escritorio: En aplicaciones de escritorio que no requieren análisis numérico o científico de datos, cargar NumPy puede ser excesivo. Puedes optar por otras librerías más ligeras si no trabajas con datos numéricos.
* Procesamiento de Texto y Cadena de Caracteres: NumPy está diseñado principalmente para operaciones numéricas y no es la elección adecuada para el procesamiento de texto o manipulación de cadenas de caracteres. En su lugar, Python ofrece herramientas nativas para trabajar con cadenas de caracteres.
* Proyectos que Requieren Mínimos Requisitos de Espacio y Recursos: En entornos con recursos muy limitados, como sistemas embebidos, NumPy puede consumir más memoria y recursos de CPU de los que están disponibles.
* Programación de Scripts Rápidos: Si estás escribiendo un script rápido o una tarea única que no involucra cálculos numéricos, la carga de NumPy puede ser innecesaria y agregar complejidad.

Se recomienda evaluar necesidades específicas del proyecto y elegir las herramientas y librerías que mejor se adapten a esas necesidades.

* ¿Qué se necesita para usar la librería Numpy?

Para usar la librería NumPy en Python, se requiere realizar los siguientes pasos:

* Instalación de NumPy: se debe tener Python instalado en tu sistema, ya que NumPy es una librería de Python
* Luego, se debe instalar NumPy.
* En tu código Python, importa NumPy utilizando la siguiente línea:

python

Copy code

import numpy as np

asi estará listo para usarse, una vez importado NumPy, se puede utilizar sus funciones y estructuras de datos en tu código. La estructura de datos principal de NumPy es el "ndarray" (arreglo multidimensional), que te permite realizar operaciones numéricas eficientes en datos multidimensionales.

* ¿Cómo se usa la librería Numpy?

La librería NumPy se utiliza principalmente para trabajar con arreglos multidimensionales (ndarrays) y realizar operaciones numéricas en ellos.

|  |  |
| --- | --- |
| 1. Importar NumPy: | importar NumPy en tu script o entorno de Python. Es común importar NumPy con el alias "np" para hacer que el código sea más legible: |
| python  Copy code  import numpy as np |
| 2. Crear Arreglos NumPy | Desde listas o tuplas de Python:  python  Copy code  arr = np.array([1, 2, 3, 4, 5])  Utilizando funciones específicas de NumPy para crear arreglos predefinidos: |
| python  Copy code  zeros\_arr = np.zeros((3, 4)) # Crea un arreglo de ceros de tamaño 3x4  ones\_arr = np.ones((2, 2)) # Crea un arreglo de unos de tamaño 2x2 |
| Generando secuencias de números:  python  Copy code  sequence\_arr = np.arange(0, 10, 2) # Crea un arreglo con valores del 0 al 10 (exclusivo) con paso de 2 |
| 3. Operaciones Numéricas | NumPy ofrece una amplia gama de operaciones matemáticas que puedes realizar en arreglos NumPy. Algunos ejemplos incluyen:  Suma, resta, multiplicación y división de arreglo  result = arr1 + arr2  result = arr1 - arr2  result = arr1 \* arr2  result = arr1 / arr2 |

**RETO**

Una compañía de ciencia de datos necesita hacer una prueba de concepto, en la que se verifique la esperanza de tratamiento de información bajo un entorno controlado, usando la librería numpy y su objeto ndarray. Por lo anterior, se solicita lo siguiente:

1. Crear un objeto de tipo ndarray, con números enteros entre 64 hasta 1024.
2. Redimensionar a una matriz bidimensional que tenga las 1024 observaciones, pero que su dimensión tenga tanto el mismo número de filas como de columnas, con la finalidad de obtener una cuadrada.
3. Crear una columna de datos aleatorios de 30 observaciones o registros, en la que se escojan dichos registros desde una lista de 10 nombres. Se debe repetir el proceso anterior para crear un set de números de cédulas aleatorio para cada uno de los nombres.

Lista=[‘Andres’,’Maria’,’Manuel’,’Daniel’,’Sarah’,’Cristian’,’Violetta’,’Lucia’,’Jackson’,’Jose’]

1. Concatenar un nuevo vector con las observaciones de los nombres y cédulas.
2. Mostrar la posición del número más grande de las cédulas generadas del vector creado en el punto 4. Se debe investigar Slicing notation en numpy.

Finalmente, implemente el código de la solución al problema usando el lenguaje de programación Python de manera que cumpla con los requerimientos que se piden y siguiendo las indicaciones que se le dan en el enunciado. Se deben tener en cuenta los detalles de documentación, y explicar lo que considere necesarios.

**Respuesta**: Se adjunta el código en archivo .py denominado **ana\_milena\_rosero\_unigarro\_reto\_7\_8.py.**