

CONSULTA LIBRERÍA NUMPY (PYTHON)

Por: Juan Daniel Torres Ramirez – 2240082

1) ¿Qué es NumPy?

- NumPy es una librería de Python para computación científica, NumPy significa Python numérico y a continuación se presentará un resumen de la descripción presentada en el sitio oficial de la librería.

“NumPy es el paquete fundamental para la computación científica en Python, es una librería que provee objetos de arreglos multidimensionales, varios objetos derivados (como arreglos y matrices enmascaradas) y una variedad de rutinas para operaciones rápidas en arreglos que incluyen: operaciones matemáticas, lógicas, de ordenamiento, de búsqueda y mucho más”.

A su vez, es importante mencionar una serie de ventajas que esta librería posee:

- ✓ Poderosos objetos de arreglos N-Dimensionales
- ✓ Herramientas para integrar código en C/C++
- ✓ Capacidades de álgebra lineal, transformaciones de Fourier y números aleatorios.

2) ¿Qué funciones proporciona para el uso de vectores y arreglos?

- Uno de los principales beneficios que posee NumPy es una generación y manejo de datos veloz, esto se debe a que esta librería tiene una estructura de datos propia llamada **arreglo** que es similar a las **listas** de Python, pero es más eficiente al almacenar y operar datos.

A continuación, se presentan algunas de las funciones que NumPy proporciona para trabajar con vectores y arreglos.

I. Creación de arreglos

- `np.array(lista)`: Crea un arreglo a partir de una lista o tupla.
- Inicialización:

- `np.zeros((filas,columnas))`: Matriz llena de ceros.
- `np.ones((filas, columnas))`: Matriz llena de unos.
- `np.eye(N)`: Matriz identidad NxN.
- `np.empty(shape)`: Arreglo sin inicializar, más rápido para grandes dimensiones.
- `np.arange(inicio, fin, paso)`: Secuencia de valores espaciados.
- `np.linspace(inicio, fin, num)`: Valores equiespaciados en un intervalo.
- `np.random.rand(filas, columnas)`: Valores aleatorios entre 0 y 1 para una matriz.
- `np.full((shape), valor)`: Arreglo lleno de un valor específico.

II. Operaciones Matemáticas

- Aritmética básica (elemento a elemento)
 - `+, -, *, /, //` (división de enteros), `**` (potencia).
 - Ejemplo: `vector * 2` (cada elemento se multiplica por 2).
- Funciones universales (ufuncs):
 - `np.sqrt()`, `np.exp()`, `np.log()`, `np.sin()`, `np.cos()`, `np.abs()`, entre otros.
- Productos
 - `np.dot(a,b)` o `a @ b`: Producto punto o escalar.
 - `np.cross(v1, v2)`: Producto cruz (Para vectores 3D).
 - `np.outer(v1,v2)`: Producto externo.
 - `np.inner(v1,v2)`: Producto interno.

III. Manipulación de formas y estructura

- Redimensionar
 - `a.reshape(n,m)`: Cambia la forma del arreglo.
 - `a.flatten()` ó `a.ravel()`: Convierte una matriz en un vector plano.
- Transposición
 - `a.T` ó `np.transpose(a)`: Transpone filas y columnas.

- Concatenación
- `np.concatenate((a, b), axis=0/1)`: Une arreglos a lo largo de un eje.
- `np.vstack((a,b))`: Apila verticalmente
- `np.hstack((a,b))`: Apila horizontalmente.
- Division:
- `np.split(a, índices)`: Divide el arreglo "a" en partes a lo largo de un eje específico (por defecto, eje 0).
- `np.vsplit(a, partes)`: División vertical (a lo largo del eje 0, filas).
- `np.hsplit(a, partes)`: División horizontal (a lo largo del eje 1, columnas).

IV. Álgebra lineal (`numpy.linalg`)

- `np.linalg.inv(a)`: Inversa de una matriz.
- `np.linalg.det(a)`: Determinante.
- `np.linalg.eig(a)`: Autovalores y autovectores.
- `np.linalg.solve(a,b)`: Resuelve el sistema lineal $ax = b$.
- `np.linalg.norm(a)`: Norma vectorial o matricial.
- `np.linalg.matrix_rank(a)`: Rango de la matriz.
- `np.linalg.qr(a)` : Descomposición QR.
- `np.linalg.svd(a)` : Descomposición en valores singulares (SVD).

V. Estadística y agregaciones

- Reducción de dimensiones
- `np.sum(a)`, `np.mean(a)`, `np.max(a)`, `np.min(a)`, `np.std(a)` [Desviación estándar].
- Uso por eje: `axis=0` (columnas) o `axis=1` (filas).
- Ejemplo: `np.mean(matriz, axis=0)` -> Promedio de cada columna.
- Índices extremos
- `np.argmax(a)`, `np.argmin(a)`: Posición del máximo/mínimo.

VI. Indexación y slicing

- Acceso basado en condiciones
- `vector[vector > 3]` #Elementos mayores a 3
- Máscaras booleanas
- `maskara = (matriz == 5)`
- `matriz[maskara] = 0` #Reemplaza valores 5 por 0

VII. Broadcasting

- Permite operar arreglos de distintas formas automáticamente
- `a = np.array([1,2], [3,4])`
- `b = np.array([10,20])`
- `a + b` #Resultado: `[[11,22] , [13,24]]`

VIII. Funciones útiles

- Ordenamiento
- `np.sort(a)`: Ordena elementos.
- `a.argsort()`: Índices que ordenan el arreglo.
- Copiado
- `np.copy(a)`: Copia profunda del arreglo.
- Atributos clave
- `a.shape`: Tupla con las dimensiones.
- `a.dtype`: Tipo de datos (ej. `Int32`, `float64`).
- `a.size`: Número total de elementos.
- `a.ndim`: Número de dimensiones.

IX. Ejemplo de uso

```
import numpy as np

# Crear matriz 3x3 y calcular su inversa
matriz = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
inversa = np.linalg.inv(matriz) # Si es invertible

# Multiplicar matriz por vector
vector = np.array([1, 0, 2])
resultado = matriz @ vector # Producto matricial
```

3) Conclusión

- NumPy es una librería muy útil y práctica que permitirá aplicar los conceptos matemáticos y lógicos vistos en materias anteriores como álgebra lineal I para resolver, adaptar y ejecutar programas de una manera mas ordenada, eficiente y profesional, con lo cual, permitirá a los estudiantes enriquecer su conocimiento a través de experiencias que los acercaran cada vez más a entornos reales de aplicaciones del desarrollo de software y, al dominarlo, se eleva no solo la calidad del código, sino que también la capacidad para abordar problemas complejos de manera estructurada y escalable.

4) Bibliografía

Alberca, A. S. (s/f). *La librería Numpy*. Aprende con Alf. Recuperado el 3 de marzo de 2025, de <https://aprendeconalf.es/docencia/python/manual/numpy/>

Cardellino, F. (2021, marzo 20). *La guía definitiva del paquete NumPy para computación científica en Python*. freecodecamp.org.

<https://www.freecodecamp.org/espanol/news/la-guia-definitiva-del-paquete-numpy-para-computacion-cientifica-en-python/>

NumPy documentation — NumPy v2.2 manual. (s/f). Numpy.org. Recuperado el 3 de marzo de 2025, de <https://numpy.org/doc/stable/>