

Tutorial: Operaciones Básicas con Matrices en Numpy

1. Creación de Matrices

- **np.array()** : Para crear una matriz, simplemente pasa una lista o tupla como argumento.

```
matriz = np.array([[1, 2], [3, 4]])
```

- **np.matrix()** : Otra forma de crear matrices, aunque **np.array()** es más recomendado.

```
matriz = np.matrix([[1, 2], [3, 4]])
```

2. Multiplicación Matricial

- `A @ B` o `np.dot(A, B)` : Para multiplicar matrices A y B matricialmente.

```
resultado = A @ B
```

- `A * B` o `np.multiply(A, B)` : Para multiplicar matrices elemento a elemento.

3. Determinante de una Matriz

- `np.det()` : Calcula el determinante de una matriz.

```
det = np.linalg.det(matriz)
```

4. Acceder a Elementos de una Matriz

- Las matrices en Numpy usan indexación basada en 0. Para acceder al elemento en la segunda fila y tercera columna:

```
elemento = matriz[1, 2]
```

5. Matrices Identidad

- `np.eye()` o `np.identity()` : Crea una matriz identidad del tamaño especificado.

```
identidad = np.eye(4)
```

6. Transpuesta de una Matriz

- `.T` o `np.transpose()` : Obtiene la transpuesta de una matriz.

```
transpuesta = matriz.T
```

7. Inversa de una Matriz

- `np.inv()` : Calcula la inversa de una matriz.

```
inversa = np.linalg.inv(matriz)
```


8. Eigenvalores y eigenvectores

- `np.eig()` : Calcula los autovalores y autovectores de una matriz.

```
eigenvals, eigenvecs = np.linalg.eig(A)
```

9. Matrices Diagonales

- `np.diag()` : Crea una matriz diagonal a partir de una lista o tupla.

```
diagonal = np.diag([2, 4, 6])
```

10. Concatenación de Matrices

- **np.vstack()** : Concatena matrices verticalmente.

```
vertical = np.vstack((matriz1, matriz2))
```

- **np.hstack()** : Concatena matrices horizontalmente.

```
horizontal = np.hstack((matriz1, matriz2))
```

Con este tutorial, deberías poder realizar operaciones básicas con matrices en Numpy. ¡Practica con ejemplos reales para afianzar tus conocimientos!