

## Matrices

Las matrices en Python son estructuras de datos que permiten almacenar una colección de elementos organizados en filas y columnas.

Una forma sencilla de representar una matriz en Python es la siguiente:

```
matriz = [[1,2,3],  
          [4,5,6],  
          [7,8,9]]
```



Filas

Columnas			
	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Cada fila estará compuesta por una lista de elementos, donde cada elemento corresponde a una columna.

## Matrices – mostrar filas

En Python podremos mostrar las filas de una matriz de la siguiente manera:

```
matriz = [[1,2,3],  
          [4,5,6],  
          [7,8,9]]  
  
for fila in matriz:  
    print(fila)
```

Muestra por pantalla



```
[1, 2, 3]  
[4, 5, 6]  
[7, 8, 9]
```

Se muestra por pantalla cada fila de la matriz, donde cada fila es una lista de elementos.

## Matrices – mostrar elementos

También podremos mostrar cada elemento de la matriz por separado:

```
matriz = [[1,2,3],  
          [4,5,6],  
          [7,8,9]]  
  
for fila in matriz:  
    for elemento in fila:  
        print(elemento)
```

Muestra por pantalla



1  
2  
3  
4  
5  
6  
7  
8  
9

Se muestra por pantalla cada elemento de la matriz de forma separada.



## Matrices – mostrar elementos con la estructura de una matriz

Además, podremos mostrar cada elemento respetando el formato que tiene la estructura de una matriz.

```
matriz = [[1,2,3],  
          [4,5,6],  
          [7,8,9]]  
  
for fila in matriz:  
    for elemento in fila:  
        print(elemento, end= " ")  
    print()
```

Muestra por pantalla



1	2	3
4	5	6
7	8	9



### Aplicaciones practicas

Las matrices son útiles en varios campos donde se requieren operaciones matemáticas o procesamiento estructurado de datos en dos o mas dimensiones. Por ejemplo:

**Matemáticas y algebra lineal** : para resolver sistemas de ecuaciones lineales , suma, resta o multiplicaciones de matrices. Matriz inversa, transpuesta, identidad, etc.

**Procesamiento de imágenes**: como una imagen es una matriz de pixeles (en escala de grises o RGB) se pueden implementar filtros, transformaciones o reconocimiento de patrones.

**Informática**: Para representar hojas de calculo y bases de datos.

**Videojuegos**: Para representar mapas, tableros o niveles de videojuegos. Por ejemplo, la posición de cada pieza en un tablero de ajedrez.

**Inteligencia Artificial**: Los datos de entrenamiento de modelos de IA se representan como matrices.