

Sesión # 13: Componente Práctico

Matrices o vectores bidimensionales

En Python podemos trabajar los arreglos bidimensionales como listas de listas, es decir, cada elemento de la lista es una lista.

Nota: Existe una librería en Python que maneja tanto vectores como matrices llamada numpy. Esta librería está por fuera del alcance de este curso, pero puedes investigarla.

Veamos un ejemplo:

```
def ejemplo1():  
    a = [[1, 2, 3, 4, 5],  
         [6, 7, 8, 9, 0]]  
    print(a)
```

```
ejemplo1()
```

O podemos recorrer todos los elementos e imprimir como una matriz

```
def ejemplo2():  
    a = [[1, 2, 3, 4, 5],  
         [6, 7, 8, 9, 0]]  
    for i in range(len(a)):  
        for j in range(len(a[i])):  
            print(a[i][j], end=' ')  
        print()
```

```
ejemplo2()
```

Actividades:

Actividad 1: Vamos a escribir una función que llene una matriz de 5 filas y 10 columnas con números enteros entre 1 y 100 aleatorios, imprima los valores máximo y mínimo y sus posiciones dentro de la matriz.

Actividad 2: El producto escalar de un número real, x , y una matriz A es la matriz xA . Cada elemento de la matriz xA es x veces su elemento correspondiente en A . Diseñemos una función `escala (matriz, escalar)` que dada `matriz[n][m]` y un escalar, imprima el producto escalar de la matriz.