

Clase 3.1

Colecciones de datos (DS o EDDs)

Una colección de datos es una estructura que guarda y ordena datos, permitiendo realizar acciones como acceder a los datos, modificar y manipular los datos. En la clase de hoy revisaremos

- ▶ Listas
- ▶ Matrices 2D o n -Dimensionales

Listas

Una lista en Python es una estructura de datos que te permite almacenar y **organizar múltiples elementos (incluso distintos tipos de datos)** en una sola variable.

```
# Lista de numeros
numeros = [1, 2, 3, 4, 5]

# Lista de cadenas de texto
nombres = ["Juan", "Maria", "Luisa"]

# Lista mixta
mixta = [1, "dos", True, 3.14]
```

Lista-Interactuando

Recorrer una Lista / Agregar Elementos al final de la lista

```
# RECORRER
```

```
numeros = [1, 2, 3, 4, 5]  
for numero in numeros:  
    print(numero)
```

```
# AGREGAR ELEMENTOS
```

```
numeros = [1, 2, 3]  
numeros.append(4)  
print(numeros)    # Output: [1, 2, 3, 4]
```

Listas-Interactuando

Insertar elementos: usa indexación tradicional (similar a range)

Eliminar: recibe el elemento como argumento

```
# INSERTAR ELEMENTOS EN UNA POSICION ESPECIFICA
```

```
numeros = [1, 2, 3, 4]
```

```
numeros.insert(2, 2.5)
```

```
print(numeros) # Output: [1, 2, 2.5, 3, 4]
```

```
# ELIMINAR ELEMENTOS
```

```
numeros = [1, 2, 3, 4]
```

```
numeros.remove(3)
```

```
print(numeros) # Output: [1, 2, 4]
```

Listas-Interactuando

Solo aplica para listas con los mismos tipos de datos, si no lanza un `TypeError`

```
# ORDENAR UNA LISTA
```

```
numeros = [3, 1, 4, 2]
```

```
numeros.sort()
```

```
print(numeros)    # Output: [1, 2, 3, 4]
```

```
# USO DE ARGUMENTO REVERSE
```

```
numeros = [3, 1, 4, 2]
```

```
numeros.sort(reverse=True)
```

```
print(numeros)    # Output: [4, 3, 2, 1]
```

Matrices 2D

Las matrices consisten en una estructura de datos rectangular (2 dimensiones) que almacenan datos en un número determinado de celdas, agrupadas en filas (horizontales) y columnas (verticales)

```
# MATRIZ DE 3 FILAS x 3 COLUMNAS
# NUMERO DE FILAS PUEDE SER 0,1,2
# NUMERO DE COLUMNAS PUEDE SER 0,1,2
matriz_sencilla = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

# PARA RECORRER
for elemento in matriz_sencilla:
    print(elemento)
```

Matrices 2D

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

```
# ACCEDER A UN ELEMENTO
```

```
print(matriz_sencilla[1][0])
```

```
# IMPRIMIR ELEMENTOS
```

```
for fila in matriz_sencilla:
```

```
    for elemento in fila:
```

```
        print(elemento, end=' ')
```


Matrices n -dimensionales

```
# Crear una matriz tridimensional (2x3x3)
arreglo_3D =
[
  [1, 2, 3],
  [4, 5, 6],
],
[
  [7, 8, 9],
  [10, 11, 12],
],
[
  [13, 14, 15],
  [16, 17, 18],
]
```

Matrices n-dimensionales

```
# Recorrer la matriz tridimensional
# e imprimir cada elemento
for x in range(len(matriz_3d)):
    for y in range(len(matriz_3d[x])):
        for z in range(len(matriz_3d[x][y])):
            print(f"Elemento
                  ({x}, {y}, {z}):
                  {matriz_3d[x][y][z]}")
```