

TC1028

Pensamiento Computacional para Ingeniería

Matrices

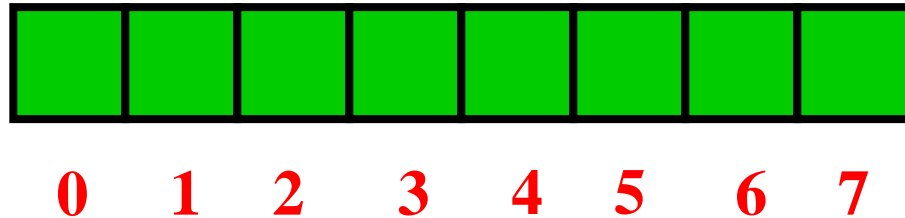
ITESM Campus Querétaro



Matriz o lista anidada

Una **lista o arreglo** lo definimos como **la relación entre un nombre y un conjunto de localidades.**

Decimos que la estructura de datos que definimos en el arreglo es de una sola dimensión ya que utilizamos un solo valor para identificar a cada localidad (0,1,2,...,n-1).



Matriz o lista anidada

Una **matriz o lista anidada** también es una colección de localidades asociadas a un nombre, sólo que los datos se organizan en **dos dimensiones**. Por ello, para hacer referencia a una localidad del arreglo se necesitan de dos números:

- * El **número de renglón**
- * El **número de columna**

	0	1	2	3	4	5
0						
1						
2						
3						

Matriz o lista anidada

- De la misma manera que en los arreglos, la numeración de renglones y de columnas inicia desde **0**.
- Si por ejemplo definimos una matriz de **6** renglones y **6** columnas, el primer elemento de la colección se encontraría en:
 - **renglón 0**
 - **columna 0**
- Y el último elemento se encontraría en:
 - **renglón 5**
 - **columna 5**



Definición de una matriz

- La definición de variables de tipo matriz es similar a los arreglos, la forma general de declarar una variable matriz es la siguiente:

matriz = [][]

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

1	2	3
4	5	6
7	8	9

Localidades de una matriz

- Una vez que hemos declarado la variable matriz, **¿Cómo tenemos acceso a los valores?** indicando el **renglón** y la **columna** de dicha localidad.
- Para esto debemos recordar que los **renglones** están numerados de **0** a **r-1** (en donde r es el número de renglones del arreglo) y que las **columnas** están numeradas de **0** a **c-1** (en donde c es el número de columnas del arreglo).

1	2	3
4	5	6
7	8	9

Localidades de una matriz

- Para hacer referencia a una localidad específica de una matriz debemos escribir el nombre de la variable y entre corchetes el número de renglón y el número de columna de la localidad.

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

- Así por ejemplo, la manera como referenciamos a la primera localidad de la matriz **M** que se definió anteriormente es **M[0][0]** y la última localidad es **M[2][2]**

1	2	3
4	5	6
7	8	9

Asignaciones a una localidad de la matriz

- La forma de asignar un valor a una localidad específica de la matriz es la siguiente:

nombre[**renglon**][**columna**] = **valor**

- en donde **nombre** es el nombre de la variable matriz, **renglon** es el número del renglón de la localidad y **columna** el número de columna de la localidad de la matriz y **valor** es cualquier valor del tipo con que fue definida la matriz.

1	2	3
4	5	6
7	8	9

Actividad Grupal

Realizar ejercicios con matrices para comprender su funcionamiento.



¿Cómo quedaría la matriz siguiente al ejecutar las instrucciones descritas?

$M = [[1,20,30,40], [2,10,20,20], [30,20,10,20], [40,20,20,10]]$

$x1 = 10$

$x2 = 20$

$M[0][0] = x1$

$M[1][0] = x2$

$M[0][3] = M[0][0] * x2$

	0	1	2	3
0	10	20	30	?
1	20	10	20	20
2	30	20	10	20
3	40	20	20	10



¿Cual sería el valor de X dada la siguiente matriz?

$M = [[10, 20, 30, 40], [20, 10, 20, 20], [30, 20, 10, 20], [40, 20, 20, 10]]$

$X = M[0][0] + M[0][1] + M[0][2] + M[0][3]$

	0	1	2	3
0	10	20	30	40
1	20	10	20	20
2	30	20	10	20
3	40	20	20	10



Matrices o listas anidadas

```
a = [[1, 2, 3], [4, 5, 6]]
```

```
print(a[0])
```

```
print(a[1])
```

El primer elemento de **a** aquí - **a[0]** - es una lista de números **[1, 2, 3]** .

El segundo elemento de **a** aquí - **a[1]** - es otra lista de números **[4, 5, 6]** .

El primer elemento de esta nueva lista es **a[0][0] == 1**; además:

```
a[0][1] == 2
```

```
a[0][2] == 3
```

```
a[1][0] == 4
```

```
a[1][1] == 5
```

```
a[1][2] == 6
```

Matrices o listas anidadas

- Para procesar una matriz bidimensional, normalmente utilizas ciclos anidados.
- El primer ciclo itera a través del número de renglón, el segundo ciclo recorre los elementos dentro de un renglón.

1	2	3
4	5	6
7	8	9

Matrices o listas anidadas

- La estructura compañera de las matrices son **dos ciclos for anidados**.
- El acceso a los elementos de una matriz puede ser de dos formas:

1. A través del iterador

```
for renglon in matriz:  
    for elemento in renglon:  
        print(elemento, end=' ')  
    print()
```

2. A través del índice

```
for i in range(0, len(matriz)):  
    for j in range(0, len(matriz[i])):  
        print(matriz[i][j], end=' ')  
    print()
```



For anidados

- Esta estructura de doble ciclo nos permite recorrer todas las localidades el arreglo, con el primer ciclo **for** se recorren los **renglones**, y con el segundo ciclo se recorren las **columnas** de cada renglón.
- En el primer ciclo se recorre cada **renglón**, para ello la variable del ciclo (**i**) toma los valores:

0, 1, 2, ... , len(matriz) – 1

que son precisamente los números de cada renglón de la matriz.

For anidados

- En el segundo ciclo para el renglón **i**, se recorre cada **columna** de la matriz, para ello la variable del ciclo (**j**) toma los valores:

0, 1, 2, ... , len(matriz[i])-1

que son precisamente los números de cada columna de la matriz.

1	2	3
4	5	6
7	8	9

Actividad Grupal

Escriba el código de la función **iniciaMatriz**, que recibe una matriz de enteros de **3** renglones y **3** columnas y le asigna a cada localidad el valor de **5**.



```
def iniciaMatriz (M):  
    for i in range(0, len(M)):  
        for j in range(0, len(M[i])):  
            M[ i ][ j ] = 5;
```

1	2	3
4	5	6
7	8	9

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

```
iniciaMatriz(matriz)
```



Solución

Actividad Grupal

Escriba el código de la función **imprimeMatriz**, que recibe una matriz de enteros de enteros de **3** renglones y **3** columnas y despliega en pantalla el contenido de la matriz.



```
def imprimeMatriz (M):
```

```
    for i in range(0, len(M)):
```

```
        for j in range(0, len(M[i])):
```

```
            print(M[ i ][ j ], end= '  ' )
```

```
        print()
```

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

```
imprimeMatriz(matriz)
```

Solución



1	2	3
4	5	6
7	8	9

Actividad

(3 minutos)

Escriba el código de la función **iniciaMatriz2**, que recibe una matriz de enteros de **3** renglones y **3** columnas y le asigna a cada localidad un número consecutivo correspondiente del **1** al **9**



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

```
def iniciaMatriz2 (M):
```

```
    num = 1
```

```
    for i in range(0, len(M)):
```

```
        for j in range(0, len(M[i])):
```

```
            M[ i ][ j ] = num
```

```
            num = num + 1
```

1	2	3
4	5	6
7	8	9

```
matriz = [ [5, 5, 5],[5, 5, 5],[5, 5, 5] ]
```

```
iniciaMatriz2(matriz)
```



Solución

Actividad

(5 minutos)



- Escriba el código de la función **sumaMatrices**, que recibe las matrices A, B y C de enteros de **3** renglones y **3** columnas cada una.
- El procedimiento asignará en la localidad correspondiente de la matriz C la suma de las matrices A más B.

$$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 9 & 10 & 11 \end{bmatrix} + \begin{bmatrix} 1 & 3 & 5 \\ 9 & 11 & 13 \\ 17 & 19 & 21 \end{bmatrix} = \begin{bmatrix} 2 & 5 & 8 \\ 14 & 17 & 20 \\ 26 & 29 & 32 \end{bmatrix}$$

```
def sumaMatrices (A, B, C):  
    for i in range(0, len(A)):  
        for j in range(0, len(A[i])):  
            C[ i ][ j ] = A[i][j] + B[i][j]
```

1	2	3
4	5	6
7	8	9

```
m1 = [ [1, 2, 3 ],[4, 5, 6],[7, 8, 9] ]
```

```
m2 = [ [5, 5, 5 ],[5, 5, 5],[5, 5, 5] ]
```

```
m3 = [ [0, 0, 0 ],[0, 0, 0],[0, 0, 0] ]
```

```
sumaMatrices(m1, m2, m3)
```



Solución

Actividad Grupal

Integrar todos los procedimientos vistos anteriormente en un solo programa para verificar su funcionamiento

