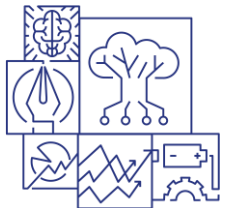


Programación Básica Clase 10

Agenda

Arreglos multidimensionales

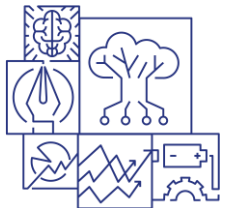
- Concepto y utilización
- Creación
- Escritura
- Ejemplos



Arreglos multi dimensionales

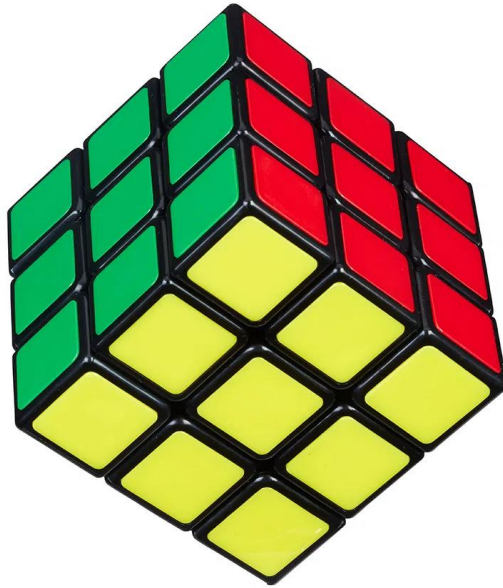


	0	1	2
0	23	54	12
1	21	81	14
2	16	20	21



Los arreglos multidimensionales son arreglos de más de una dimensión.

Para efectos de este curso se estudiarán únicamente los arreglos bidimensionales, también nos referiremos a ellos como matrices.



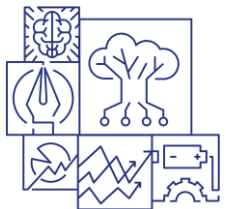
Un arreglo bidimensional está conformado por filas y columnas. Cada fila tiene un número asociado, así también cada columna.

Por lo cual, cada elemento en el arreglo puede ser referenciado por dos índices, uno indica la fila y otro la columna.

En la siguiente figura se almacenan las puntuaciones obtenidas por tres jugadores en un torneo realizado durante tres días.

The diagram shows a 3x3 matrix with rows labeled 0, 1, 2 and columns labeled 0, 1, 2. A red box labeled 'Filas' points to the row indices, and a red box labeled 'Columnas' points to the column indices. A green box labeled 'matriz[1][2]' points to the value 14 in the second row and third column.

	0	1	2
0	23	54	12
1	21	81	14
2	16	20	21



Para representar la situación anterior, en Python podríamos declarar un arreglo de la siguiente forma:

```
puntosJuego=[[23,54,12],[21,81,14],[16,20,21]]
```

Como se puede observar en el ejemplo, se crea un arreglo de dos dimensiones de 3 x 3 elementos. Cada grupo de elementos encerrados entre corchetes corresponde a una fila del arreglo.

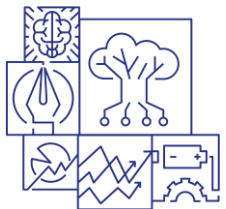


Asignar un valor al arreglo bidimensional

Gráficamente, se muestra que el número 54 está siendo agregado a la matriz.

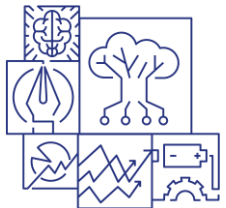
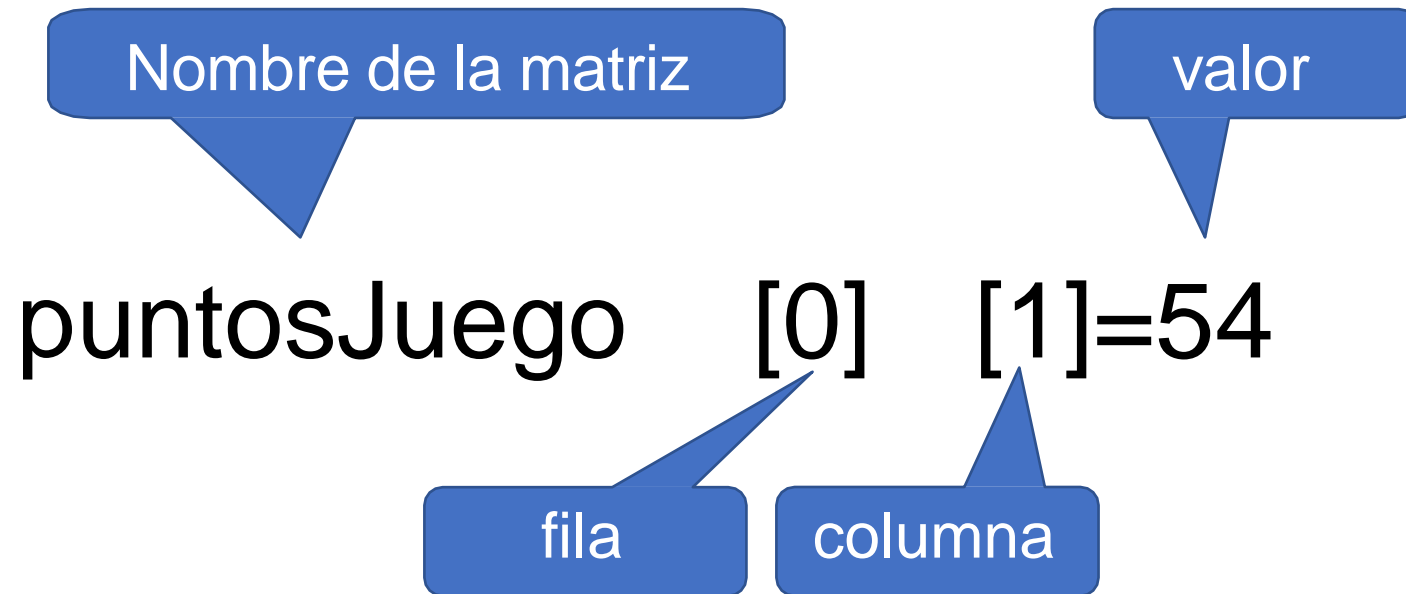
	0	1	2
0	23	54	12
1	21	81	14
2	16	20	21

`puntosJuego[0][1]=54`



Para asignar un valor al arreglo bidimensional, se debe hacer referencia al nombre del arreglo seguido de la posición fila y columna, entre corchetes, seguido del símbolo de igual (=) y finalmente el valor a asignar.

Siguiendo el ejemplo anterior en Python lo representamos de la siguiente manera:



Agregando datos a un arreglo bidimensional

A través de la utilización de dos ciclos accedemos a cada una de las posiciones del arreglo bidimensional. Siguiendo con el ejemplo anterior quedaría de la siguiente manera:

```
puntosJuego=[[0,0,0],[0,0,0],[0,0,0]]
for x in range(0,3):
    for y in range(0,3):
        valor=int(input("Digite la puntuación obtenida:"))
        puntosJuego[x][y]=valor
```

Declaración o creación del arreglo

Asigna el valor que indicó el usuario



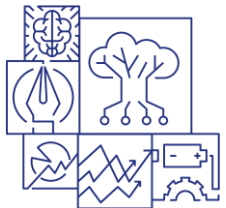
Imprimir los datos del arreglo bidimensional

Así como lo hicimos para almacenar datos en el arreglo anterior, para mostrar los datos que contiene, utilizamos dos ciclos, de la siguiente manera:

```
for x in range(0,3):  
    for y in range(0,3):  
        print(puntosJuego[x][y],end=" ")  
    print("\n")
```

Dentro de la función print junto al dato a imprimir se agrega un delimitador o separador (end=" "), el cual crea un espacio entre cada dato.

Además, se usa el carácter \n que es un salto de línea e indica dónde termina una fila de modo que la impresión de la información continúe en la siguiente.

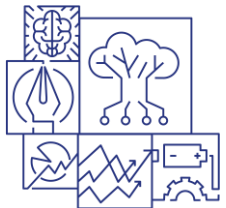


Al imprimir la matriz quedaría así:

23 54 12

21 81 14

16 20 21



Modificar un valor en el arreglo bidimensional

En algunas ocasiones será necesario modificar valores almacenados. Dada la siguiente matriz:

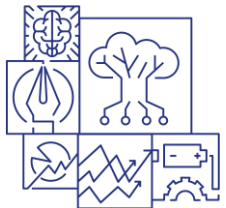
23 54 12

21 81 14

16 20 21

Supongamos que deseamos penalizar con 5 puntos al jugador que tiene 14 puntos por una falta cometida durante el desarrollo del torneo. Para ello, debemos acceder al tercer valor de la segunda fila (14). Procedemos de la siguiente forma:

```
puntosJuego[1][2]=puntosJuego[1][2]-5
```



Ejercicio

1



Comparta con su profesor, algunos casos en los que se considera que es necesario utilizar un arreglo de dos dimensiones para almacenar los datos.

De los casos mencionados anteriormente, desarrolle un ejercicio relacionado, su profesor podrá orientarlo al respecto.



Tiempo aproximado 30 minutos



Ejercicio

2



Un profesor necesita un programa que calcule la **nota final** de sus estudiantes. Tiene 25 estudiantes que realizan 4 actividades evaluativas (para efectos de esta clase, se reducirá a 5 estudiantes).

Utilice una matriz para almacenar las calificaciones donde cada fila representará un estudiante y las columnas almacenarán la información de las actividades evaluativas.

Tiempo aproximado 30 minutos



Ejercicio

3



Desarrolle un programa para reservar espacios en una microbús que da servicio en 4 horarios. Se le pide que inicialmente almacene un valor 0 en los 20 espacios disponibles, luego le solicite al usuario la posición que desea reservar, remplazando el valor de 0 por un 1 (que representa vendido).

¿Qué otro valor se podría utilizar?

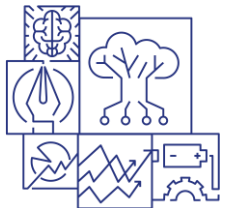
Nota: Recuerde que el índice de las filas y columnas empieza en cero pero para el usuario la fila empieza en 1.

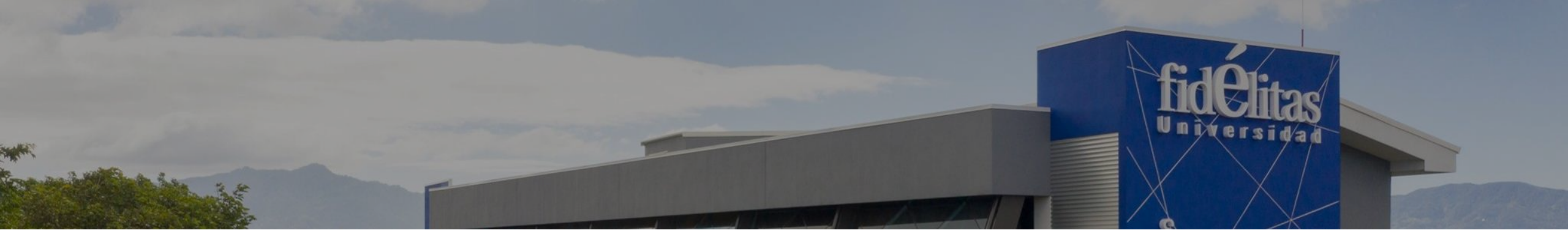


Tiempo aproximado 40 minutos



Existen arreglos con mayor cantidad de dimensiones, son de mucha utilidad cuando se tienen más categorizaciones para los datos, pero están fuera del alcance de este curso.





¡Nos vemos la próxima semana!

