Arreglos Bidimensionales

Lógica y Algoritmia


Imagen tomada de: http://goo.gl/VstJw2
Arreglos Bidimensionales

Arreglos Bidimensionales (1)

- Son un conjunto finito homogéneo y ordenado de elementos a los cuales se accede utilizando dos índices, uno para las filas y otro para las columnas.

Índice	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Arreglos Bidimensionales (2)

- "Un arreglo de arreglos unidimensionales" es una manera de interpretar a los arreglos bidimensionales:

Índice	**0**	**1**	**2**
Arreglo 0	1	2	3

Índice	**0**	**1**	**2**
Arreglo 1	4	5	6

Índice	**0**	**1**	**2**
Arreglo 2	7	8	9

Arreglos Bidimensionales (3)

- Para utilizarlos se requiere de una librería para Python llamada “NumPy”. Para utilizarla se debe incluir la siguiente instrucción al inicio de cada programa:

```
import numpy as np
```

- Se accede a sus funciones a través del objeto creado, en este caso “np”

Arreglos Bidimensionales (4)

- La primera forma de crear un arreglo bidimensional definiendo el número de filas y de columnas rellenándolo de ceros con la función de numpy “zeros”.

Índice	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0

3 filas y 3 columnas

Arreglos Bidimensionales: Ejemplo 1 (1) crear arreglo

Código:

```
import numpy as np
x=np.zeros(shape=(3,3))
print(x)
```

Ejecución:

```
>>>
[[ 0.  0.  0.]
 [ 0.  0.  0.]
 [ 0.  0.  0.]]
>>>
```

Arreglos Bidimensionales (5)

- La segunda forma de crear un arreglo bidimensional es definiendo directamente el contenido y tamaño del mismo con la función de numpy “array”.

Índice	0	1	2
0	1	2	3
1	2	4	6
2	3	6	9

3 filas y 3 columnas

Arreglos Bidimensionales: Ejemplo 1 (2) crear arreglo

Código:

```
import numpy as np
x=np.array([[1,2,3],[2,4,6],[3,6,9]])
print(x)
```

Ejecución:

```
>>>
[[1 2 3]
 [2 4 6]
 [3 6 9]]
>>>
```

Arreglos Bidimensionales (6)

- Para leer o modificar un dato del arreglo se hace uso de los índices de filas y columnas.

Índice	0	1	2
0	1	2	3
1	2	4	6
2	3	6	9

Leer dato de la fila 2 columna 1

Arreglos Bidimensionales: Ejemplo 2 (1) leer dato del arreglo

Código:

```
import numpy as np
x=np.array([[1,2,3],[2,4,6],[3,6,9]])
print(x[2,1])
```

Ejecución:

```
>>>
6
>>>
```

Arreglos Bidimensionales: Ejemplo 2 (2) Modificar dato

Código:

```
import numpy as np
x=np.array([[1,2,3],[2,4,6],[3,6,9]])
print("Anterior:")
print(x[2,1])
x[2,1]=8
print("Nuevo:")
print(x[2,1])
```

Ejecución:

```
>>>
Anterior:
6
Nuevo:
8
>>>
```

Arreglos Bidimensionales (7)

- Para saber la cantidad de elementos de un arreglo bidimensional se utiliza la función de numpy “size”.

Índice	0	1	2
0	1	2	3
1	2	4	6
2	3	6	9

La cantidad de elementos en el arreglo es de 9.

Arreglos Bidimensionales: Ejemplo 3 cantidad de elementos

Código:

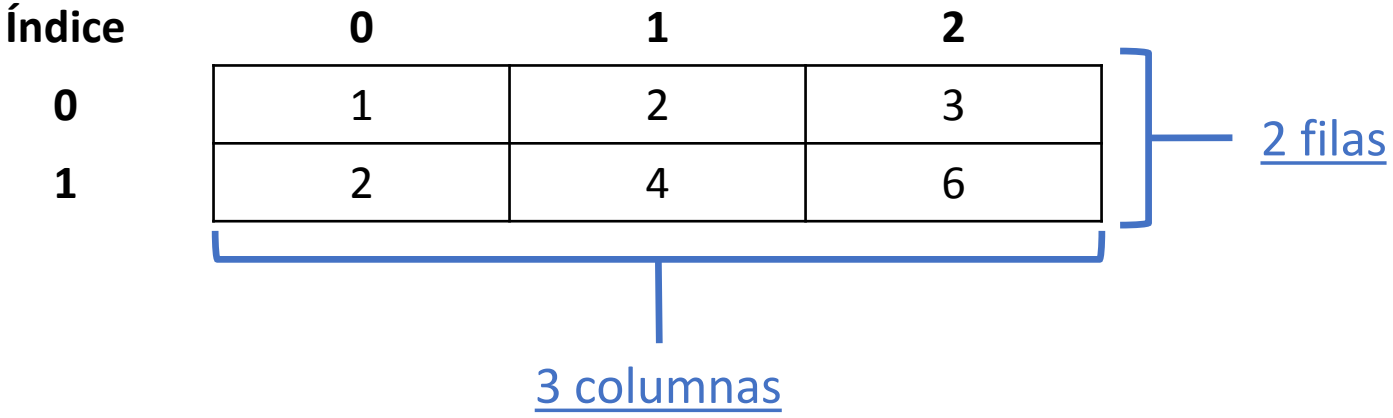
```
import numpy as np
x=np.array([[1,2,3],[2,4,6],[3,6,9]])
print(x.size)
```

Ejecución:

```
>>>
9
>>>
```

Arreglos Bidimensionales (8)

- Para conocer la cantidad de filas y de columnas que posee el arreglo bidimensional se utiliza la función de numpy “shape”.



Arreglos Bidimensionales: Ejemplo 4 filas y columnas

Código:

```
import numpy as np
x=np.array([[1,2,3],[2,4,6]])
print(x.shape)
```

Ejecución:

```
>>>
(2, 3)
>>>
```

Arreglos Bidimensionales (9)

- El ordenamiento de arreglos bidimensionales utilizando la función de numpy “sort” ordena únicamente los valores de cada uno de los arreglos unidimensionales que contiene.

Desordenados:

Índice	**0**	**1**	**2**
0	6	3	9
1	4	2	6

Ordenados:

Índice	**0**	**1**	**2**
0	3	6	9
1	2	4	6

Arreglos Bidimensionales: Ejemplo 5 sort

Código:

```
import numpy as np
x=np.array([[6,9,3],[6,2,4]])
print("Antes:")
print(x)
x=np.sort(x)
print("Después:")
print(x)
```

Ejecución:

```
>>>
Antes:
[[6 9 3]
 [6 2 4]]
Después:
[[3 6 9]
 [2 4 6]]
>>>
```

Arreglos Bidimensionales (10)

- Utilizando el ciclo repetitivo “for” se pueden imprimir uno por uno los arreglos unidimensionales del arreglo bidimensional.

Arreglo completo:

Índice	**0**	**1**	**2**
0	1	2	3
1	4	5	6

Arreglo separado:

Índice	**0**	**1**	**2**
Arreglo 0	1	2	3

Índice	**0**	**1**	**2**
Arreglo 1	4	5	6

Arreglos Bidimensionales: Ejemplo 6 arreglos separados

Código:

```
import numpy as np
x=np.array([[1,2,3],[4,5,6]])
for arreglo in x:
    print("Arreglo ",arreglo)
```

Ejecución:

```
>>>
Arreglo  [1 2 3]
Arreglo  [4 5 6]
>>>
```

Arreglos Bidimensionales (11)

- Para leer los datos uno por uno del arreglo bidimensional se sigue la lógica anterior utilizando los ciclos “for” anidados.

Arreglo completo:

Índice	0	1	2
0	3	6	9
1	2	4	6

Arreglo separado:

Arreglo 1					
Dato 1:	3	Dato 2:	6	Dato 3:	9
Arreglo 2					
Dato 4:	2	Dato 5:	4	Dato 6:	6

Arreglos Bidimensionales: Ejemplo 7 datos separados

Código:

```
import numpy as np
x=np.array([[3,6,9],[2,4,6]])
for arreglo in x:
    for dato in arreglo:
        print(dato)
```

Ejecución:

```
>>>
3
6
9
2
4
6
>>>
```

Referencias Bibliográficas

[1] Zelle, John M. Python Programming an Introduction to Computer Science. -- 2nd ed. -- Washington : Franklin, Beedle & Associates Inc, 2010.

[2] Dawson, Michael. Python Programming for the Absolute Beginner. -- 3th ed. -- Australia : Cengage Learning, 2010.

[3] Rosaura Gutiérrez Almeyda, Urbano Eliécer Gómez Prada, Jairo Viola, y Diana Teresa Gómez Forero. Presentación de clase: Arreglos Bidimensionales en Python con NumPy. Universidad Pontificia Bolivariana Seccional Bucaramanga. 2015.