

Arreglos Unidimensionales

Lógica y Algoritmia



Arreglos Unidimensionales

Arreglos Unidimensionales (1)

- Son una colección finita de elementos homogéneos identificados por un índice.

Índice	0	1	2	3	4
Dato	1	2	3	4	5

Índice	0	1	2	3	4
Dato	'a'	'b'	'c'	'd'	'e'

Índice	0	1	2	3	4
Dato	1.0	1.5	2.0	2.5	3.0

Arreglos Unidimensionales (2)

- 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 Unidimensionales (3)

- Estas son algunas de las funciones de NumPy:
 - Array: Crear un arreglo con valores definidos.
 - Zeros: Crea un arreglo de n espacios y lo llena con ceros.
 - Size: Indica la cantidad de elementos en el arreglo.
 - Sort: Permite realizar un ordenamiento a los elementos del arreglo.

Arreglos Unidimensionales (4)

- Función “array” de NumPy permite crear arreglos definiendo directamente el tamaño y contenido del mismo:

```
a = np.array([1, 2, 3, 4, 5])
```

Índice	0	1	2	3	4
Dato	1	2	3	4	5

Arreglos Unidimensionales:

Ejemplo 1 crear arreglo

Código:

```
import numpy as np
a = np.array([1,2,3,4,5])
print (a)
```

Ejecución:

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

Arreglos Unidimensionales (5)

- Se pueden leer y escribir en el arreglo directamente utilizando los índices o posiciones:

	0	1	2	3	4
Dato	1	2	3	4	5

Reemplazar dato en el índice 2:



	0	1	2	3	4
Dato	1	2	9	4	5



El número 9 fue ingresado en la posición 2 del arreglo:

Arreglos Unidimensionales: Ejemplo 2 (1) lectura por índice

Código:

```
import numpy as np  
a = np.array([1, 2, 3, 4, 5])  
print(a[0])
```

Ejecución:

```
>>>  
1  
>>>
```

Arreglos Unidimensionales: Ejemplo 2 (2) cambiar dato

Código:

```
import numpy as np
a = np.array([1,2,3,4,5])
a[2]=9
print(a)
```

Ejecución:

```
>>>
[1 2 9 4 5]
>>>
```

Arreglos Unidimensionales (6)

- La función “zeros” de NumPy permite definir el tamaño del arreglo y llenarlo con ceros:

```
a = np.zeros(5)
```

Índice	0	1	2	3	4
Dato	0.0	0.0	0.0	0.0	0.0

Arreglos Unidimensionales: Ejemplo 3 “zeros”

Código:

```
import numpy as np  
a = np.zeros(5)  
print(a)
```

Ejecución:


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

Arreglos Unidimensionales (7)

- Función “size” de NumPy:

```
a = np.array([1, 2, 3, 4, 5])  
print(a.size)
```

Índice	0	1	2	3	4
Dato	0.0	0.0	0.0	0.0	0.0



Imprime la cantidad de elementos en el arreglo, en este caso 5

Arreglos Unidimensionales: Ejemplo 4 “size”

Código:

```
import numpy as np  
a = np.array([1,2,3,4,5])  
print(a.size)
```

Ejecución:

```
>>>  
5  
>>>
```

Arreglos Unidimensionales (8)

- Los arreglos con datos numéricos (enteros o decimales) pueden ser ordenadas de manera ascendente con la función “sort”.

Índice	0	1	2	3	4
Dato	2.5	3	4.5	5	1

Índice	0	1	2	3	4
Dato	1	1.5	2	2.5	3

Arreglos Unidimensionales: Ejemplo 5 “sort”

Código:

```
import numpy as np
a = np.array([2, 5, 1, 3, 4])
a = np.sort(a)
print(a)
```


Ejecución:

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


Arreglos Unidimensionales (9)

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

Índice	0	1	2	3	4
Dato	1	2	3	4	5



Imprimir todos los datos de la lista uno por uno

Arreglos Unidimensionales: Ejemplo 6 imprimir con ciclo “for”

Código:

```
import numpy as np
a = np.array([1,2,3,4,5])
for elemento in a:
    print(elemento)
```

Ejecución:

```
>>>
1
2
3
4
5
>>>
```

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 Unidimensionales en Python con NumPy. Universidad Pontificia Bolivariana Seccional Bucaramanga. 2015.