El array es el principal objeto de la librería. Representa datos de manera estructurada y se puede acceder a ellos a través del indexado, a un dato específico o un grupo de muchos datos específicos.

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

lista

---> [1, 2, 3, 4, 5, 6, 7, 8, 9]

Volvemos nuestra lista, un array

arr = np.array(lista)

arr

---> [1, 2, 3, 4, 5, 6, 7, 8, 9]

Una matriz son varios ***Vectores*** o ***listas*** agrupadas una encima de la otra, es como una tabla de Excel

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

matriz = np.array(matriz)

matriz

---> array([[1, 2, 3],

[4, 5, 6],

[7, 8, 9]])

El ***indexado*** nos permite acceder a los elementos de los array y matrices  
Los elementos se ***empiezan a contar desde 0.***

arr[0]

---> 1

| **Index** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Es posible ***operar*** directamente con los elementos.

arr[0] + arr[5]

---> 7

| **0** | **+** | **5** |
| --- | --- | --- |
| 1 |  | 6 |

En el caso de las ***matrices***, al indexar una posición se regresa el array de dicha posición.

matriz[0]

---> array([1, 2, 3])

| **Index** | **0** | **1** | **2** |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| 1 | 4 | 5 | 6 |
| 2 | 7 | 8 | 9 |

Para seleccionar un solo elemento de la matriz se especifica la posición del elemento ***separada por comas.***

**Nota:** El primer elemento selecciona las filas, el segundo elemento las columnas

matriz[0, 2]

---> 3

**Slicing**

Nos permite extraer varios datos, tiene un comienzo y un final.  
En este ejemplo se está extrayendo datos desde la posición 1 hasta la 5.

arr[1:6]

---> array([2, 3, 4, 5, 6])

Si no se ingresa el ***valor de inicio***, se toma el inicio como la posición 0.

arr[:6]

---> array([1, 2, 3, 4, 5, 6])

En cambio, si no se le da una ***posición final***, se regresan todos los elementos hasta el final del array.

arr[2:]

---> array([3, 4, 5, 6, 7, 8, 9])

También se puede ***trabajar por pasos.***

En este ejemplo de 3 en 3.  
Regresa la posición 0, 0 + 3, 3 + 3 y como no hay posición 6 + 3, no se regrese nada.

arr[::3]

---> array([1, 4, 7])

| **Index** | **0** | **3** | **7** |
| --- | --- | --- | --- |
| 0 | 1 | 4 | 7 |

Cuando se le asigna un ***valor negativo*** se regresan los valores comenzando desde la última posición del array.

arr[-1]

---> 9

arr[-3:]

---> array([7, 8, 9])

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |

Para el caso de las matrices, sucede algo similar.  
Para acceder a los valores entre filas.

matriz[1:]

---> array([[4, 5, 6],

[7, 8, 9]])

Para acceder a los valores entre filas y columnas.

matriz[1:, 0:2]

---> array([[4, 5],

[7, 8]])

**Reto**