



Numpy

---





Es un paquete de Python que proporciona herramientas para operar con matrices multidimensionales. Permite realizar una gran variedad de rutinas para operaciones rápidas en matrices, incluida la manipulación matemática, lógica, de formas, clasificación, selección, transformadas discretas de Fourier, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria y mucho más.

# Introducción a NumPy



```
>>> a[(0,1,2,3,4),(1,2,3,4,5)]  
array([ 1, 12, 23, 34, 45])
```

```
>>> a[3:,[0, 2, 5]]  
array([[30, 32, 35],  
       [40, 42, 45]],  
      [50, 52, 55])
```

```
>>> mask = array([1,0,1,0,0,1],  
                 dtype=bool)
```

```
>>> a[mask,2]  
array([2,22,52])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

# Introducción a NumPy

- Numeric Python – Computación Científica
- Objeto **numpy-array** (arreglo n-dimensional)
- Se importa con el siguiente comando:

```
import numpy as np
```

- Llamar a una función de numpy tiene la siguiente estructura:

```
np.function(args)
```



# Arrays con NumPy

- Array es una matriz unidimensional.
- En Python la estructura de datos por defecto son las listas. Sin embargo, las listas no soportan operaciones matemáticas avanzadas, no está optimizado para hacerlo.
- NumPy = Paquete de Python creado por Travis Oliphant con propósitos científicos.
- Array en NumPy consume menos memoria y por ende es más rápido que una lista de Python.

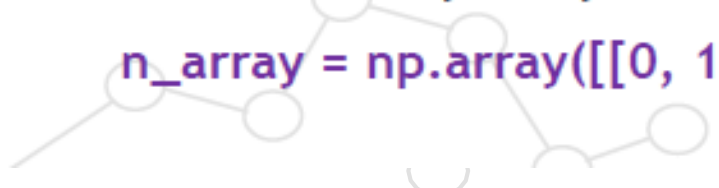
## *Para la creación de un Array en Python*

1. Necesitamos importar la librería:

```
import numpy as np
```

2. Creamos el array en Python:

```
n_array = np.array([[0, 1, 2, 3],[4, 5, 6, 7],[8, 9, 10, 11]])
```



# Arrays con NumPy

*Un Array en Python tiene los siguientes atributos:*

- `ndim`: nos da el número de dimensiones del array.

`n_array.ndim`

- `shape`: nos da el tamaño de cada dimensión.

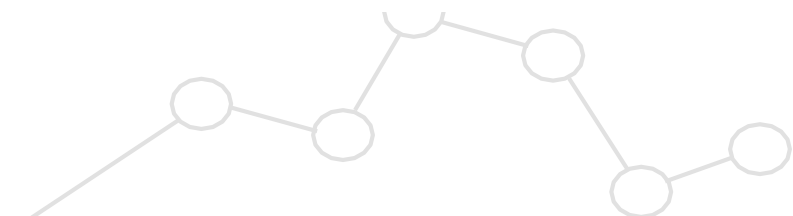
`n_array.shape`

- `size`: nos da el número de elementos.

`n_array.size`

- `dtype`: nos da el tipo de los datos en el array.

`n_array.dtype.name`



# Arrays con NumPy

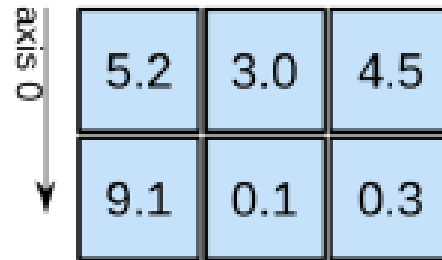
1D array



axis 0 →

shape: (4,)

2D array

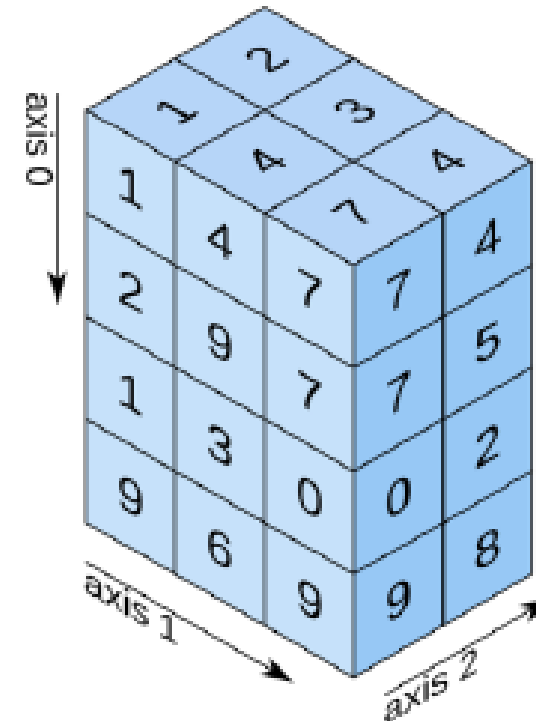


axis 0 ↓

axis 1 →

shape: (2, 3)

3D array



shape: (4, 3, 2)

# Operaciones Matemáticas con NumPy

- Restas de arrays.

```
a = np.array( [11, 12, 13, 14])
```

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

```
c = a - b
```

```
c
```

- Potencia de arrays.

```
b**2
```

- Funciones sobre los arrays.

```
np.cos(b)
```

- Multiplicación de matrices

```
A1 = np.array([[1, 1],[0, 1]]) / A2 = np.array([[2, 0],[3, 4]])
```

```
A1*A2
```

```
np.dot(A1,A2)
```





# Indexing and Slicing

*Si deseamos seleccionar un elemento en particular de un array:*

- Seleccionamos la fila 1 y la columna 2.

`n_array[0,1]`

- Seleccionamos la fila 1 y las 3 primeras columnas.

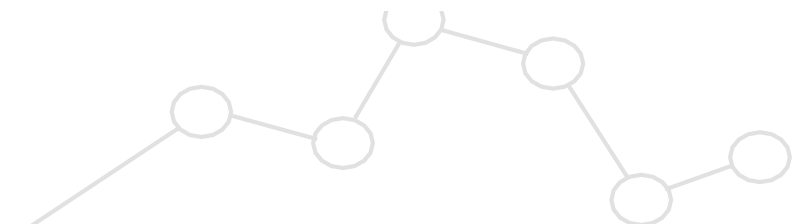
`n_array[ 0 , 0:3 ]`

- Seleccionamos la fila 1 y todas las columnas.

`n_array[ 0 , : ]`

- Seleccionamos todas las filas y la columna 2.

`n_array[ : , 1 ]`



# Indexing y Slicing

```
>>> a[0,3:5]  
array([3,4])
```

```
>>> a[4:,4:]  
array([[44, 45],  
       [54, 55]])
```

```
>>> a[:,2]  
array([2,12,22,32,42,52])
```

```
>>> a[2::2,::2]  
array([[20,22,24],  
       [40,42,44]])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55