

# Propiedades y manipulación de arrays.

Téllez Gerardo Rubén

2/5/2021

## Array plano

No solo nos da los objetos, tiene propiedades que permite a la librería trabajar rápidamente

```
import numpy as np
x = np.arange(0, 12)
print(x)
```

```
## [ 0  1  2  3  4  5  6  7  8  9 10 11]
```

```
x1 = x.reshape((3,4))
print(x1)
```

```
## [[ 0  1  2  3]
##   [ 4  5  6  7]
##   [ 8  9 10 11]]
```

En python, al número de dimensiones se le llama rango

## Matriz

Pedir número de dimensiones

```
print(x1.ndim)
```

```
## 2
```

Dimensiones de la matriz

```
print(x1.shape)
```

```
## (3, 4)
```

Solicitar tipo de dato

```
#print(x1.dtype)
```

Solicitar los bits del objeto

```
#print(x1.itemsize)
```

Solicitar la ubicación de memoria

```
print(x1.data)
```

```
## <memory at 0x00000000220717E0>
```

## Codificar atributos

```
print(x1)
```

```
## [[ 0  1  2  3]
##   [ 4  5  6  7]
##   [ 8  9 10 11]]
```

```
x2 = x1
x2.shape = (4,3)
print(x2)
```

```
## [[ 0  1  2]
##   [ 3  4  5]
##   [ 6  7  8]
##   [ 9 10 11]]
```

## Llamar submatrices

```
xsub = x2[1:3, 1:3]
print(xsub)
```

```
## [[4 5]
##   [7 8]]
```

## Llamar subectores

```
print(x1[0:5])
```

```
## [[ 0  1  2]
##   [ 3  4  5]
##   [ 6  7  8]
##   [ 9 10 11]]
```

```
za = np.arange(0, 21)
print(za)
```

```
## [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
```

```
zb = np.arange(10, 6, -1)
print(za[zb])
```

```
## [10  9  8  7]
```

## Llamar por condición booleana

```
print(za[(za <= 10)])
```

```
## [ 0  1  2  3  4  5  6  7  8  9 10]
```

## Asignar valores a los elementos seleccionados

El tipo de dato debe respetarse, NumPy va a transformar el tipo de datos de ser posible, de lo contrario arroja un error.

```
za1 = za
za1[(za1 < 10)] = 10
print(za1)
```

```
## [10 10 10 10 10 10 10 10 10 10 10 11 12 13 14 15 16 17 18 19 20]
```

```
za1[0:6] = 5
print(za1)
```

```
## [ 5  5  5  5  5  5 10 10 10 10 10 11 12 13 14 15 16 17 18 19 20]
```