

Creado por:

Isabel Maniega

## Ejercicio 1

Crea una matriz con ayuda numpy:

**1)** Cuya matriz contenga 4 filas por 4 columnas de ceros

```
In [2]: import numpy as np
```

```
In [3]: matriz = np.zeros((4,4))  
matriz
```

```
Out[3]: array([[0., 0., 0., 0.],  
              [0., 0., 0., 0.],  
              [0., 0., 0., 0.],  
              [0., 0., 0., 0.]])
```

**2)** Apartir de la matriz de ceros crea la matriz identidad



```
In [6]: for i in range(len(matriz)):  
        # print(i)  
        for j in range(len(matriz[i])):  
            # print('Valor de j: ', j)  
            if j == i:  
                matriz[i][j] = 1  
matriz
```

```
Out[6]: array([[1., 0., 0., 0.],  
              [0., 1., 0., 0.],  
              [0., 0., 1., 0.],  
              [0., 0., 0., 1.]])
```

## Ejercicio 2

Crea una matriz con ayuda numpy:

- Primera fila contenga: 3, 6, 8
- Segunda fila contenga: 20, 5, 7
- Tercera fila contenga: 10, 14, 1

```
In [7]: m = np.array([  
        [3, 6, 8],  
        [20, 5, 7],  
        [10, 14, 1]
```

```
] )  
m
```

```
Out[7]: array([[ 3,  6,  8],  
              [20,  5,  7],  
              [10, 14,  1]])
```

**1) Crea la transpuesta de esta matriz**

```
In [8]: m.T
```

```
Out[8]: array([[ 3, 20, 10],  
              [ 6,  5, 14],  
              [ 8,  7,  1]])
```

**2) Muestra el tamaño de la matriz**

```
In [10]: m.size
```

```
Out[10]: 9
```

**3) Muestra las dimensiones**

```
In [11]: m.shape
```

```
Out[11]: (3, 3)
```

**4) ¿La matriz tiene valores menores de 0?**

```
In [13]: np.all(m<0)
```

```
Out[13]: False
```

**5) ¿La matriz tiene algún valor mayor de 10?**

```
In [14]: np.any(m>10)
```

```
Out[14]: True
```

**6) Coge una muestra de 5 valores de esa matriz usando linspace**

```
In [18]: muestra = np.linspace(m.min(), m.max(), 5)  
muestra
```

```
Out[18]: array([ 1. ,  5.75, 10.5 , 15.25, 20. ])
```

**7) La matriz contiene el valor 7**

```
In [19]: 7 in m
```

```
Out[19]: True
```

**8) Crea una copia de esa matriz en una única dimensión**

```
In [21]: m.flatten()
```

```
Out[21]: array([ 3,  6,  8, 20,  5,  7, 10, 14,  1])
```

**9)** Crea una copia de esa matriz en una única dimensión, en este caso usando un bucle y una lista vacía

```
In [25]: lista_final = []
        for i in range(len(m)):
            for j in range(len(m[i])):
                lista_final.append(m[i][j])
        print(lista_final)
        m2 = np.array(lista_final)
        m2
```

```
[3, 6, 8, 20, 5, 7, 10, 14, 1]
```

```
Out[25]: array([ 3,  6,  8, 20,  5,  7, 10, 14,  1])
```

**10)** Realiza a esta última matriz creada con flatten la potencia de 3

```
In [26]: pow(m2, 3)
```

```
Out[26]: array([ 27,  216,  512, 8000,  125,  343, 1000, 2744,  1])
```

*Creado por:*

*Isabel Maniega*