

code002

June 4, 2024

```
[ ]: #Segunda practica con Numpy  
#Importando Numpy  
import numpy as np
```

```
[ ]: #Trabajando con arrays y sus distintas propiedades  
arreglo_1 = np.array([1,2,3,4])  
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
```

Arreglo = [1 2 3 4]
Tipo de datos = int64

```
[ ]: #Cambiendo el tipo de dato del arreglo  
arreglo_1 = np.array([1,2,3,4], dtype='float64')  
#Dentro del mismo array definimos el tipo de dato  
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
```

Arreglo = [1. 2. 3. 4.]
Tipo de datos = float64

```
[ ]: #Cambiendo el tipo de dato ya definido  
#Definiendo como entero  
arreglo_1 = np.array([0,1,2,3,4])  
arreglo_1 = arreglo_1.astype(np.float64)  
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
```

Arreglo = [0. 1. 2. 3. 4.]
Tipo de datos = float64

```
[ ]: #Recordando el tipo de dato anterior  
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
```

Arreglo = [0. 1. 2. 3. 4.]
Tipo de datos = float64

```
Arreglo = [False True True True True]
Tipo de datos = bool
```

```
[ ]: #Re-declarando el array
arreglo_1 = np.array([0,1,2,3,4])
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
#Cambiando a string el arreglo1
arreglo_1 = arreglo_1.astype(np.string_)
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
```

```
Arreglo = [0 1 2 3 4]
Tipo de datos = int64
Arreglo = [b'0' b'1' b'2' b'3' b'4']
Tipo de datos = |S21
```

```
[ ]: #Haciendo el proceso inverso
arreglo_1 = np.array(['0','1','2','3','4'])
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
arreglo_1 = arreglo_1.astype(np.int8)
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}')
```

```
Arreglo = ['0' '1' '2' '3' '4']
Tipo de datos = <U1
Arreglo = [0 1 2 3 4]
Tipo de datos = int8
```

```
[ ]: #Haciendo que falle la conversion debido a que no existe una correspondencia
entre 'hola' y un número
```

```
arreglo_1 = np.array(['0','hola','1','2','3','4'])
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}\n')
arreglo_1 = arreglo_1.astype(np.int8)
print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}\n')
```

```
Arreglo = ['0' 'hola' '1' '2' '3' '4']
Tipo de datos = <U4
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[18], line 5
      3 arreglo_1 = np.array(['0','hola','1','2','3','4'])
      4 print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}\n')
----> 5 arreglo_1 = arreglo_1.astype(np.int8)
      6 print(f'Arreglo = {arreglo_1}\nTipo de datos = {arreglo_1.dtype}\n')

ValueError: invalid literal for int() with base 10: 'hola'
```

Es importante resaltar que Numpy solo puede manejar un tipo de datos dentro de los arreglos, ya que es parte de la optimización de datos