

m8-julihocastillo

March 9, 2024

1 Módulo 8 | Arrays

¡Bienvenidos a la actividad práctica del módulo!

Antes de empezar

En este ejercicio, desarrollarás funciones utilizando las técnicas que hemos visto, con el objetivo de resolver los problemas detallados aquí.

Entregable: Un Jupyter Notebook (archivo de extensión .ipynb), archivo PDF y capturas de pantallas en espacio de respuesta que muestren tanto el código desarrollado como la solución al problema planteado, incluyendo los comentarios que sean pertinentes a las preguntas que se plantean. Entregas sin estos elementos no serán calificadas.

¡Buena suerte!

Paso a paso:

1.1 Ejercicio 1

Escribe un programa que mediante indexación en NumPy sea capaz de invertir un arreglo (donde el primer elemento se convierte en el último)

```
[ ]: import numpy as np

def invertir_arreglo(arreglo):
    return arreglo[::-1]

# Crear un arreglo de ejemplo
arreglo_original = np.array([1, 2, 3, 4, 5])

# Usar la función para invertir el arreglo
arreglo_invertido = invertir_arreglo(arreglo_original)

print(arreglo_invertido)
```

```
[5 4 3 2 1]
```

```
[ ]: arr = np.array([12,13,14,15,16,17,18,19,20])
      invertir_arreglo(arr)
```

```
[ ]: array([20, 19, 18, 17, 16, 15, 14, 13, 12])
```

1.2 Ejercicio 2

Escribe un programa con NumPy para convertir valores de grados Fahrenheit a grados centígrados. Los valores se almacenan en un arreglo de salida.

```
[ ]: import numpy as np

def fahrenheit_a_celsius(grados_fahrenheit):
    return (grados_fahrenheit - 32) * 5/9

# Arreglo de ejemplo con temperaturas en grados Fahrenheit
temperaturas_fahrenheit = np.array([32, 212, 0, 100, 68])

# Convertir a Celsius
temperaturas_celsius = fahrenheit_a_celsius(temperaturas_fahrenheit)

print(temperaturas_celsius)

[ 0.          100.         -17.77777778  37.77777778  20.          ]
```

```
[ ]: arr = np.array([0, 12, 45.21, 34, 99.91, 32])
fahrenheit_a_celsius(arr)
```

```
[ ]: array([-17.77777778, -11.11111111,  7.33888889,  1.11111111,
          37.72777778,  0.          ])
```

2 Ejercicio 3

Escribe un programa con NumPy para encontrar la unión de dos arreglos. La unión debe devolver un arreglo ordenado de valores que están en cualquiera de los dos arreglos de entrada (sin repetición).

```
[ ]: import numpy as np

def union_arreglos(arreglo1, arreglo2):
    # Concatenar los dos arreglos
    arreglos_concatenados = np.concatenate((arreglo1, arreglo2))
    # Encontrar elementos únicos y ordenarlos
    return np.sort(np.unique(arreglos_concatenados))

# Definir dos arreglos de ejemplo
arreglo1 = np.array([1, 2, 3, 4, 5])
arreglo2 = np.array([3, 4, 5, 6, 7])

# Encontrar la unión de los dos arreglos de manera funcional
union = union_arreglos(arreglo1, arreglo2)
```

```
print(union)
```

```
[1 2 3 4 5 6 7]
```

```
[ ]: arr1 = np.array([0,10,20])  
arr2 = np.array([20,10,30])  
union_arreglos(arr1,arr2)
```

```
[ ]: array([ 0, 10, 20, 30])
```