

Ejercicios evaluables UD4

[Ejercicios obligatorios](#)

[Material que detrás entregar](#)

Ejercicios obligatorios

Ejercicio 1. Dadas las siguientes listas:

```
países = ["Alemania", "Francia", "Italia", "España", "Países Bajos", "Polonia",  
"Suecia", "Bélgica", "Austria", "Portugal"]
```

```
poblaciones = [83, 68, 59, 47, 17, 38, 10, 11, 9, 10] # En millones de  
habitantes
```

Crea un programa que muestre por pantalla:

- La población media de dichos países.
- El nombre del país menos poblado.

Ejercicio 2. Crea una función que dadas una lista de números y un número n determine si todos los números de la lista son mayores o iguales a n . La función deberá devolver True o False.

Ejercicio 3. Dada una lista con 2^n elementos, por ejemplo:

```
tenistas_top_8 = [  
    "Jannik Sinner",    # 11,330 puntos  
    "Alexander Zverev", # 8,135 puntos  
    "Carlos Alcaraz",   # 7,410 puntos  
    "Taylor Fritz",     # 4,900 puntos  
    "Casper Ruud",      # 4,480 puntos  
    "Daniil Medvedev",  # 3,930 puntos  
    "Novak Djokovic",   # 3,900 puntos  
    "Álex de Miñaur"    # 3,735 puntos  
]
```

Crear una función que imprima por pantalla el emparejamiento aleatorio de todos los elementos de la lista, por ejemplo para la anterior:

ENFRENTAMIENTOS:		
Taylor Fritz	vs	Jannik Sinner
Carlos Alcaraz	vs	Alexander Zverev
Álex de Miñaur	vs	Casper Ruud
Novak Djokovic	vs	Daniil Medvedev

NOTA: USAR SOLO FUNCIONES ESTUDIADAS

NOTA2: la lista original no debe ser modificada

Ejercicio 4. Crea una función que reciba por parámetro una matriz. La función deberá devolver una lista de dos elementos: el primero de ellos con el número de filas de la matriz y el segundo con el número de columnas de la matriz. No hagas comprobaciones de que lo que le entra sea o no una matriz, supón que siempre recibirá por parámetro una matriz válida.

Ejercicio 5. Dada una matriz como la siguiente (columna 2 son el número de habitantes y la 4 el área):

```
países = [  
    ["Argentina", "Buenos Aires", 46000000, "América", 2780400],  
    ["España", "Madrid", 48000000, "Europa", 505990],  
    ["Japón", "Tokio", 125000000, "Asia", 377975],  
    ["Sudáfrica", "Pretoria", 60000000, "África", 1221037],  
    ["Australia", "Canberra", 26000000, "Oceanía", 7692024],  
    ["Canadá", "Ottawa", 38000000, "América", 9984670]  
]
```

Crea una función que recibido un continente y la lista de países devuelva el país con mayor densidad de población (población/área).

Ejercicio 6. Crea la función `suma_matrices(m1:list, m2:list)` que dadas dos matrices `m1` y `m2` devuelva su suma. [Tutorial de cómo sumar matrices.](#)

Material que detrás entregar

Cinco ficheros 1.py, 2.py, ..., 6.py todos ellos dentro de una carpeta llamada `<nombre>_ud4` donde `<nombre>` es tu nombre.

La carpeta se comprimirá en el fichero `<nombre>_ud4.zip` que será el que entregues.

Cualquier fichero que no siga estrictamente la estructura expuesta tendrá automáticamente una calificación de no apto.