

Ejercicio 1: Limpiar texto eliminando múltiples espacios

Dado el texto:

```
texto = "Este es un texto con muchos espacios."
```

Reemplaza todos los grupos de **2 o más espacios** por un solo espacio.

Ejercicio 2: Filtrar solo líneas que contienen números

Dado:

```
lineas = ["ID: 123", "Nombre: Ana", "Edad: 24", "Ciudad: Madrid", "Código X"]
```

Filtra solo las líneas que contengan **al menos un dígito**.

Ejercicio 3: Normalizar fechas a formato YYYY-MM-DD

Dado el texto:

```
texto = "Fechas: 12/05/2022, 3-7-21, 09/1/2023"
```

- Extrae las fechas en formatos DD/MM/YYYY o D-M-YY.
- Normalízalas a un formato estándar YYYY-MM-DD usando re.sub() y grupos.

Ejercicio 4: Limpiar teléfonos dejando solo dígitos

Dado:

```
telefonos = ["+34 912-33-55-77", "(01) 234 5678", "999-888-777"]
```

- Convierte cada cadena en un teléfono compuesto **solo por números**.
- Usa re.sub(r"\D", "", telefono).

Ejercicio 5: Extraer valores monetarios

Dado:

```
texto = "Ventas: $1200, €950, $300. Total €1500."
```

Extrae todos los valores monetarios junto con su símbolo (\$ o €).

Ejercicio 6: Detectar registros con formato incorrecto

Supón que tienes datos estilo "campo: valor":

```
datos = ["edad: 32", "peso: 70kg", "altura-180", "ciudad: Madrid"]
```

Filtra los registros que **NO** cumplen el formato:

palabra: valor

Ejercicio 7: Limpiar etiquetas HTML básicas

Dado:

```
html = "<p>Hola <b>mundo</b>, esto es <i>Python</i></p>"
```

Elimina todas las etiquetas HTML.

Ejercicio 8: Extraer identificadores tipo “ID1234”

Dado el texto:

```
texto = "Productos: ID1234, id5678, REF9012, ID0001."
```

Extrae solo los identificadores que empiecen por **ID** (mayúscula o minúscula) y sigan con 4 dígitos.

Ejercicio 9: Normalizar nombres quitando caracteres especiales

Dado:

```
nombres = ["Ana López", "José-Pérez", "María#Rivas", "Luís$Gómez"]
```

Limpia cada nombre dejando **solo letras y espacios**.