Ficheros

## **Ejercicios Ficheros**

Recuerda marcar como directorio de trabajo el script donde estés ejecutando el fichero. También puedes crear una subcarpeta “archivos” donde meter todos los ficheros, para ello puedes usar rutas relativas: "./archivos/nombre\_archivo.txt"

### **Ejercicio 1: Leer un fichero de texto**

**Objetivo**: Crea un script que lea el contenido de un fichero llamado poema.txt y muestre su contenido en la consola.

1. Abre el fichero en modo lectura ("r").
2. Lee todo el contenido del fichero.
3. Muestra el contenido en pantalla.

**Pistas**:

* Usa la función open() para abrir el fichero.
* Usa el método .read() para leer todo el contenido de una sola vez.

### **Ejercicio 2: Escribir en un fichero de texto**

**Objetivo**: Crea un script que escriba un mensaje en un fichero llamado notas.txt.

1. Abre el fichero en modo escritura ("w").
2. Si el fichero no existe, debe crearse automáticamente.
3. Escribe en el fichero: "Esta es mi primera nota en este archivo."

**Pistas**:

* Recuerda que el modo "w" sobrescribe el contenido si el fichero ya existe.
* Usa el método .write() para escribir en el fichero.

### **Ejercicio 3: Añadir contenido a un fichero**

**Objetivo**: Modifica el script del **Ejercicio 2** para que no sobrescriba el contenido, sino que añada texto al final del fichero.

1. Abre el fichero notas.txt en modo añadir ("a").
2. Añade el siguiente mensaje: "Esta es otra línea añadida al archivo."

**Pistas**:

* Usa el modo "a" para añadir contenido al final del fichero sin eliminar lo anterior.

### **Ejercicio 4: Leer líneas de un fichero**

**Objetivo**: Crea un script que lea y muestre cada línea de un fichero de texto, línea por línea.

1. Usa el fichero poema.txt del **Ejercicio 1**.
2. Abre el fichero en modo lectura.
3. Lee el contenido línea por línea y muestra cada línea por separado.

**Pistas**:

* Usa un bucle for para recorrer cada línea del fichero.

### **Ejercicio 5: Manejo de excepciones al leer un fichero**

**Objetivo**: Crea un script que intente leer un fichero llamado archivo\_inexistente.txt y capture el error si el fichero no existe.

1. Usa un bloque try-except para manejar el error FileNotFoundError.
2. Si el fichero no se encuentra, muestra el mensaje "Error: El archivo no existe.".

**Pistas**:

* Usa try-except para manejar la excepción FileNotFoundError.

### **Ejercicio 6: Leer y escribir en un fichero CSV**

**Objetivo**: Crea un script que lea un fichero CSV llamado alumnos.csv, muestre su contenido y luego añada un nuevo alumno.

1. Usa el fichero alumnos.csv con las columnas: Nombre, Edad, Email.
2. Muestra el contenido de cada fila del fichero CSV.
3. Añade un nuevo alumno al fichero: "Carlos, 24, carlos@mail.com".

**Pistas**:

* Usa el módulo csv para manejar el fichero.
* Usa el método .writerow() para añadir nuevas filas.

### **Ejercicio 7: Contar líneas y palabras en un fichero**

**Objetivo**: Crea un script que cuente el número de líneas y el número total de palabras en un fichero llamado texto.txt.

1. Abre el fichero en modo lectura.
2. Recorre cada línea del fichero, contando el número de líneas.
3. Para cada línea, cuenta cuántas palabras contiene y suma el total de palabras.
4. Muestra el número total de líneas y el número total de palabras.

**Pistas**:

* Usa el método .split() para dividir una línea en palabras.
* Utiliza un contador para sumar las palabras.

### **Ejercicio 8: Buscar y eliminar duplicados en un fichero CSV**

**Objetivo**: Crea un script que lea un fichero CSV llamado clientes.csv y elimine los registros duplicados basados en la columna Email.

1. Abre el fichero en modo lectura.
2. Almacena los registros en un diccionario usando el Email como clave para eliminar duplicados.
3. Escribe los registros únicos en un nuevo fichero CSV llamado clientes\_unicos.csv.

**Pistas**:

* Usa un diccionario para evitar duplicados.
* Escribe los registros únicos en un nuevo CSV.

### **Ejercicio 9: Leer un fichero y buscar una palabra**

**Objetivo**: Crea un script que lea un fichero llamado libro.txt y busque una palabra específica introducida por el usuario.

1. Solicita al usuario que introduzca la palabra a buscar.
2. Abre el fichero y recorre su contenido.
3. Muestra cuántas veces aparece la palabra en el fichero.

**Pistas**:

* Usa el método .count() para contar las apariciones de la palabra en cada línea.

