

# MANEJO DE ARCHIVOS CON PYTHON



## Introducción al manejo de archivos de texto en Python

### Introducción

En esta lección aprenderás a crear y manipular archivos de texto en Python utilizando el método `open()`. Verás cómo escribir datos, manejar excepciones y cerrar archivos correctamente para garantizar un manejo seguro y profesional. 📄 🔧

---

### ✅ Paso 1: Crear el archivo principal para la práctica

📄 Ruta y nombre del archivo:  
`manejo-archivos.py`

## Paso 2: Abrir un archivo de texto desde Python

### Descripción:

Se abre o crea un archivo de texto llamado `prueba.txt` usando el modo de escritura "w" y se gestiona dentro de un bloque `try`.

### Código desde `manejo-archivos.py`:

```
try:
    archivo = open('prueba.txt', 'w')
except Exception as e:
    print(e)
finally:
    archivo.close()
```

### Explicación:

- Se abre o crea `prueba.txt` en modo escritura ('w').
- Si ocurre un error, se captura con `except`.
- El archivo se cierra con `finally`, asegurando que se libere el recurso.

## Paso 3: Escribir texto dentro del archivo

### Descripción:

Se usa el método `.write()` para agregar líneas de texto al archivo creado.

### Código desde `manejo-archivos.py`:

```
try:
    archivo = open('prueba.txt', 'w')
    archivo.write('Agregamos informacion al archivo\n')
    archivo.write('Adios\n')
except Exception as e:
    print(e)
finally:
    archivo.close()
```

### Explicación:

- Se escriben dos líneas en el archivo.
- El `\n` asegura que las líneas estén separadas.
- El archivo se cierra correctamente al final.

---

## Paso 4: Verifica dónde se crea el archivo

### Descripción:

El archivo `prueba.txt` se crea automáticamente en la misma carpeta donde se ejecuta el archivo Python.

### Explicación:

Al no especificar una ruta completa, el archivo se guarda en el directorio actual del proyecto. Puedes abrirlo desde el explorador del proyecto para verificar su contenido.



---

## Conclusión


En esta lección aprendiste a trabajar con archivos de texto en Python.  
Vimos cómo:

- Abrir un archivo con `open()`.
- Escribir con `.write()`.
- Usar `try`, `except` y `finally` para controlar errores y cerrar el archivo de forma segura.

Además, aprendiste a agregar saltos de línea y localizar el archivo en tu carpeta de proyecto.  

¡Bien hecho! A partir de aquí podrás manejar archivos de texto de forma segura y profesional en tus programas Python.  

---

Sigue adelante con tu aprendizaje  , ¡el esfuerzo vale la pena!

¡Saludos! 

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)