



# Manejo de Archivos en Python

Python permite **leer, escribir y manipular archivos** de forma sencilla usando la función `open()`.

---

## Abrir un Archivo, `open()`

Para trabajar con archivos, usamos `open(nombre_archivo, modo)`.

### Modos de apertura

Modo	Descripción
"r"	Lectura (modo predeterminado). Da error si el archivo no existe.
"w"	Escritura. <b>Sobrescribe</b> el archivo si existe, o lo crea si no.
"a"	Agregar al final del archivo sin borrar su contenido.
"x"	Crea un archivo nuevo. Da error si ya existe.
"b"	Modo binario (Ej: imágenes, PDF). Se usa con rb, wb, etc.

## Lectura de Archivos, `read ()`

Para **leer** un archivo, usamos "r".

Ejemplo: Leer **todo** el contenido

```
archivo = open("datos.txt", "r") # Abrir en modo Lectura
contenido = archivo.read() # Leer todo el archivo
print(contenido)
archivo.close() # Cerrar el archivo
```

Ejemplo: Leer **línea** por línea

```
archivo = open("datos.txt", "r")
for linea in archivo:
    print(linea.strip()) # ^.strip()^ quita saltos de línea extras
archivo.close()
```





Ejemplo: `readlines()` para obtener una lista

```
archivo = open("datos.txt", "r")
lineas = archivo.readlines() # Retorna una lista con todas las líneas
print(lineas)
archivo.close()
```

## Escritura en Archivos, `write()`

Para **escribir** en un archivo, usamos "w" o "a".

Ejemplo: **Sobrescribir** un archivo ("w")

```
archivo = open("nuevo.txt", "w") # Crear o sobrescribir
archivo.write("Hola, este es un nuevo archivo.\n")
archivo.write("Segunda línea de texto.")
archivo.close()
```

**Importante:** Si el archivo existe, **se borra su contenido** antes de escribir.

## Agregar texto sin borrar ("a")

```
archivo = open("nuevo.txt", "a") # Modo agregar
archivo.write("\nNueva línea añadida.")
archivo.close()
```

## Leer y Escribir con `with open()` (Recomendado)

El uso de `with open()` es más seguro porque **cierra automáticamente el archivo**.

Ejemplo: Leer un archivo

```
with open("datos.txt", "r") as archivo:
    contenido = archivo.read()
    print(contenido) # No necesitamos `close()`
```

Ejemplo: Escribir un archivo

```
with open("nuevo.txt", "w") as archivo:
    archivo.write("Este es otro archivo creado con 'with open()'")
```





## Manipulación de Archivos

Python tiene el módulo os para manipular archivos y carpetas.

### 📌 Comprobar si un archivo existe

```
import os

if os.path.exists("datos.txt"):
    print("El archivo existe.")
else:
    print("El archivo no existe.")
```

### 📌 Renombrar un archivo

```
os.rename("datos.txt", "informacion.txt")
```

### 📌 Eliminar un archivo

```
os.remove("informacion.txt")
```

### 📌 Crear una carpeta

```
os.mkdir("nueva_carpeta")
```

### 📌 Eliminar una carpeta vacía

```
os.rmdir("nueva_carpeta")
```

### 📌 Listar archivos en un directorio

```
archivos = os.listdir(".") # Lista los archivos en la carpeta actual
print(archivos)
```



## Resumen

Acción	Ejemplo
Abrir archivo	<code>open("archivo.txt", "r")</code>
Leer todo	<code>archivo.read()</code>
Leer línea por línea	<code>archivo.readline()</code>
Leer como lista	<code>archivo.readlines()</code>
Escribir (sobrescribir)	<code>open("archivo.txt", "w")</code>
Agregar texto	<code>open("archivo.txt", "a")</code>
Uso recomendado	<code>with open("archivo.txt", "r") as archivo:</code>
Comprobar existencia	<code>os.path.exists("archivo.txt")</code>
Renombrar archivo	<code>os.rename("archivo.txt", "nuevo.txt")</code>
Eliminar archivo	<code>os.remove("archivo.txt")</code>

## Automatización de Tareas con Python

Python es excelente para **automatizar tareas repetitivas**, lo que ahorra tiempo y reduce errores. Desde renombrar archivos hasta enviar correos electrónicos o extraer datos de páginas web, las posibilidades son muchas.

### ❖ ¿Qué se Puede Automatizar con Python?

Aquí hay algunos ejemplos de tareas comunes que puedes automatizar:

- Manipulación de archivos** (mover, copiar, renombrar, eliminar).
- Lectura y escritura en hojas de Excel o CSV.**
- Envío de correos electrónicos automáticos.**
- Automatización de navegadores web** (rellenar formularios, extraer datos).
- Descarga de archivos desde Internet.**
- Tareas repetitivas en el sistema operativo** (abrir programas, ejecutar comandos).





## Ejemplo 1: Renombrar Archivos en una Carpeta

Este script cambia el nombre de todos los archivos .txt en una carpeta, agregándoles un prefijo nuevo\_.

```
import os

carpeta = "C:/Users/Usuario/Documentos"

for archivo in os.listdir(carpeta):
    if archivo.endswith(".txt"):
        nuevo_nombre = "nuevo_" + archivo
        os.rename(os.path.join(carpeta, archivo), os.path.join(carpeta, nuevo_nombre))

print("Archivos renombrados correctamente.")
```

Útil para organizar archivos en una carpeta con un nuevo formato de nombres.

## Ejemplo 2: Enviar un Correo Electrónico Automático

Este script envía un email con smtplib y email.

```
import smtplib
from email.message import EmailMessage

correo_emisor = "tuemail@gmail.com"
correo_receptor = "destino@gmail.com"
contraseña = "tu_contraseña" # Se recomienda usar variables de entorno para seguridad

msg = EmailMessage()
msg["Subject"] = "Correo Automático con Python"
msg["From"] = correo_emisor
msg["To"] = correo_receptor
msg.set_content("Hola, este es un mensaje enviado automáticamente con Python.")

with smtplib.SMTP_SSL("smtp.gmail.com", 465) as servidor:
    servidor.login(correo_emisor, contraseña)
    servidor.send_message(msg)

print("Correo enviado exitosamente.")
```

Útil para enviar recordatorios, informes diarios, o mensajes automáticos.





### Ejemplo 3: Descargar Archivos desde Internet

Este script descarga un archivo de una URL.

```
import requests

url = "https://example.com/archivo.pdf"
respuesta = requests.get(url)

with open("archivo_descargado.pdf", "wb") as archivo:
    archivo.write(respuesta.content)

print("Archivo descargado correctamente.")
```

Útil para descargar documentos o imágenes automáticamente.

### Ejemplo 4: Extraer Datos de una Página Web (Web Scraping)

Este script usa [BeautifulSoup](#) para extraer títulos de una página web.

```
import requests
from bs4 import BeautifulSoup

url = "https://news.ycombinator.com/"
respuesta = requests.get(url)
soup = BeautifulSoup(respuesta.text, "html.parser")

titulos = soup.find_all("a", class_="titlelink")

for titulo in titulos:
    print(titulo.text)
```

Útil para extraer noticias, precios de productos o información de una web.





## Ejemplo 5: Automatizar Tareas del Sistema Operativo

Este script abre un programa automáticamente.

```
import os

os.system("notepad") # Abre el Bloc de notas en Windows
```

## Ejemplo 6: Programar la Ejecución de un Script con `schedule`

Este script ejecuta una función cada día a las 9:00 AM.

```
import schedule
import time

def tarea():
    print("Ejecutando tarea programada...")

schedule.every().day.at("09:00").do(tarea)

while True:
    schedule.run_pending()
    time.sleep(60)
```

Útil para ejecutar tareas automáticamente sin intervención manual.

## Resumen

Tarea Automatizada	Módulos Utilizados
Renombrar archivos	os
Enviar correos electrónicos	smtplib, email
Descargar archivos	requests
Extraer datos de una web	requests, BeautifulSoup
Abrir programas automáticamente	os
Programar ejecución de tareas	schedule

