

🐍 Python Cheatsheet – Working with Files

Curso: Google IT Automation with Python

Módulo: Programming with Files

Tema: Lectura, escritura y manejo de archivos y directorios

[Introduction] Introducción

En Python, los archivos se manejan mediante funciones integradas y módulos estándar como `os`, `os.path`, `shutil` y `csv`.

El flujo básico para trabajar con archivos es:

1. **Abrir** el archivo (`open()`).
2. **Leer o escribir** información.
3. **Cerrar** el archivo (`close()` o usar `with` para hacerlo automáticamente).

[Reading Files] Reading Files

Abrir y leer archivos

```
# Abrir archivo en modo lectura
file = open("archivo.txt", "r")

# Leer todo el contenido
content = file.read()
print(content)

# Cerrar el archivo
file.close()
```

⚠ Siempre cerrá los archivos después de usarlos, o usá `with` para que Python lo haga por vos.

Ler línea por línea

```
with open("archivo.txt", "r") as f:
    for line in f:
        print(line.strip()) # strip() elimina saltos de línea
```

- `read()` → lee todo el contenido.
- `readline()` → lee una línea a la vez.
- `readlines()` → devuelve una lista con todas las líneas.

🔄 Iterating Through Files

Podés recorrer archivos grandes sin cargarlos completamente en memoria:

```
with open("log.txt") as log:  
    for line in log:  
        process_line(line)
```

También se pueden combinar varios archivos:

```
for filename in ["data1.txt", "data2.txt"]:  
    with open(filename) as f:  
        print(f"Contenido de {filename}:")  
        print(f.read())
```

✍ Writing Files

Modos de apertura

Modo	Descripción
'r'	Lectura (por defecto)
'w'	Escritura (sobrescribe)
'a'	Agregar contenido al final
'x'	Crear archivo nuevo (falla si ya existe)

```
# Escribir contenido  
with open("nuevo.txt", "w") as f:  
    f.write("Hola mundo\n")  
  
# Agregar contenido  
with open("nuevo.txt", "a") as f:  
    f.write("Otra línea\n")
```

📁 File Paths

Tipos de rutas

- **Absoluta:** indica toda la ubicación desde la raíz. Ej: `/home/usuario/documento.txt`
- **Relativa:** depende del directorio actual. Ej: `../archivo.txt`

Separadores

SO	Ejemplo	Separador
Windows	C:\\\\Users\\\\Soy\\\\file.txt	\\
Linux / Mac	/home/soy/file.txt	/

Usá `os.path.join()` para escribir rutas multiplataforma:

```
import os

ruta = os.path.join("carpeta", "archivo.txt")
print(ruta)
```

⚙️ Working with Files

Operaciones comunes con `os` y `shutil`:

```
import os, shutil

os.rename("viejo.txt", "nuevo.txt")      # Renombrar
os.remove("archivo.txt")                  # Borrar
shutil.move("file.txt", "backup/")       # Mover archivo
os.chmod("script.py", 0o755)             # Cambiar permisos
```

📋 More File Information

```
import os

stats = os.stat("archivo.txt")
print(stats.st_size)    # Tamaño en bytes
print(stats.st_mtime)   # Última modificación
```

📁 Directories

```
import os

os.mkdir("nueva_carpeta")      # Crear directorio
os.rmdir("nueva_carpeta")      # Eliminar (vacío)
os.listdir(".")                 # Listar contenido del directorio actual

# Recorrer directorios
for root, dirs, files in os.walk("."):
    pass
```

```
print("Directorio:", root)
print("Subcarpetas:", dirs)
print("Archivos:", files)
```

CSV Files

¿Qué es un CSV?

Archivo de texto con valores separados por comas (Comma-Separated Values). Usado para intercambiar datos tabulares entre sistemas.

Ejemplo:

```
name,age,city
Ana,28,Buenos Aires
Luis,35,Córdoba
```

Reading CSV Files

```
import csv

with open("data.csv") as f:
    reader = csv.reader(f)
    for row in reader:
        print(row)
```

Generating CSV Files

```
import csv

with open("output.csv", "w", newline="") as f:
    writer = csv.writer(f)
    writer.writerow(["nombre", "edad", "ciudad"])
    writer.writerow(["Ana", 28, "Buenos Aires"])
```

Reading and Writing CSV Files with Dictionaries

```
import csv

# Leer con diccionarios
```

```
with open("data.csv") as f:
    reader = csv.DictReader(f)
    for row in reader:
        print(row["name"], row["city"])

# Escribir con diccionarios
with open("newdata.csv", "w", newline="") as f:
    campos = ["name", "age", "city"]
    writer = csv.DictWriter(f, fieldnames=campos)
    writer.writeheader()
    writer.writerow({"name": "Juan", "age": 32, "city": "Mendoza"})
```

Module Wrap-up: Managing Files with Python

- Lectura y escritura de archivos con `open()` y `with`.
- Rutas de archivo multiplataforma con `os.path`.
- Operaciones de archivos y carpetas con `os` y `shutil`.
- Acceso a metadatos y permisos con `os.stat()`.
- Lectura y escritura de archivos CSV con `csv` y `DictReader/DictWriter`.

💡 Consejo: siempre usá `with open()` para evitar fugas de recursos y manejar errores de manera segura.

Referencia oficial:

- Documentación de Python: [open\(\)](#)
- Módulo `os`
- Módulo `csv`

¿Querés que el siguiente módulo (por ejemplo, **Regular Expressions**, **Testing in Python**, o **Object-Oriented Programming**) te lo prepare ya en este mismo formato para ir armando tu carpeta de cheatsheets?