

📊 Automating Real-World Tasks with Python – Cheatsheet

Módulo 3: Automatic Output Generation

⌚ Introducción

En este módulo se aprende a **generar reportes y documentos automáticamente** usando Python. Incluye formatos comunes como **CSV, Excel, PDF, y resúmenes automatizados** para envío o análisis.

📁 Herramientas principales

Librería	Uso
csv	Leer y escribir archivos CSV
openpyxl	Manipular archivos Excel (.xlsx)
pandas	Procesar y generar datos tabulares
reportlab	Generar PDFs desde Python
docx (python-docx)	Crear y editar documentos Word
os / pathlib	Gestionar rutas y archivos de salida

Instalación rápida

```
pip install openpyxl pandas reportlab python-docx
```

🌐 Conceptos clave

◆ CSV Files

```
import csv

# Escribir CSV
with open("reporte.csv", "w", newline="") as f:
    writer = csv.writer(f)
    writer.writerow(["Nombre", "Edad", "Ciudad"])
    writer.writerow(["Juan", 30, "Buenos Aires"])

# Leer CSV
with open("reporte.csv") as f:
```

```
reader = csv.reader(f)
for fila in reader:
    print(fila)
```

◆ Excel Files

```
from openpyxl import Workbook, load_workbook

# Crear Excel
wb = Workbook()
ws = wb.active
ws.title = "Datos"
ws.append(["Nombre", "Edad", "Ciudad"])
ws.append(["Ana", 25, "Madrid"])
wb.save("reporte.xlsx")

# Leer Excel
wb = load_workbook("reporte.xlsx")
ws = wb.active
for row in ws.iter_rows(values_only=True):
    print(row)
```

◆ DataFrames con pandas

```
import pandas as pd

df = pd.DataFrame([
    {"Nombre": "Juan", "Edad": 30},
    {"Nombre": "Ana", "Edad": 25}
])
df.to_csv("reporte.csv", index=False)
df.to_excel("reporte.xlsx", index=False)
```

◆ Generar PDFs

```
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas

c = canvas.Canvas("reporte.pdf", pagesize=letter)
c.drawString(100, 750, "Reporte de Ventas")
c.drawString(100, 730, "Nombre: Juan, Total: $150")
c.save()
```

◆ Word Documents

```
from docx import Document

doc = Document()
doc.add_heading("Reporte de Empleados", level=1)
doc.add_paragraph("Nombre: Ana, Edad: 25")
doc.save("reporte.docx")
```

↻ Buenas prácticas

- Separar **código de generación de datos** del **código de exportación**.
- Usar **plantillas** para PDFs y Word cuando los reportes son repetitivos.
- Automatizar generación de múltiples formatos desde un mismo DataFrame.
- Validar rutas de salida usando `os.path` o `pathlib` para compatibilidad multiplataforma.
- Manejar **errores de permisos o archivos abiertos** para no interrumpir la automatización.

📎 Referencias

- [Python CSV Module](#)
- [openpyxl Documentation](#)
- [pandas Documentation](#)
- [ReportLab User Guide](#)
- [python-docx Documentation](#)