

CURSO DE PYTHON DESDE CERO



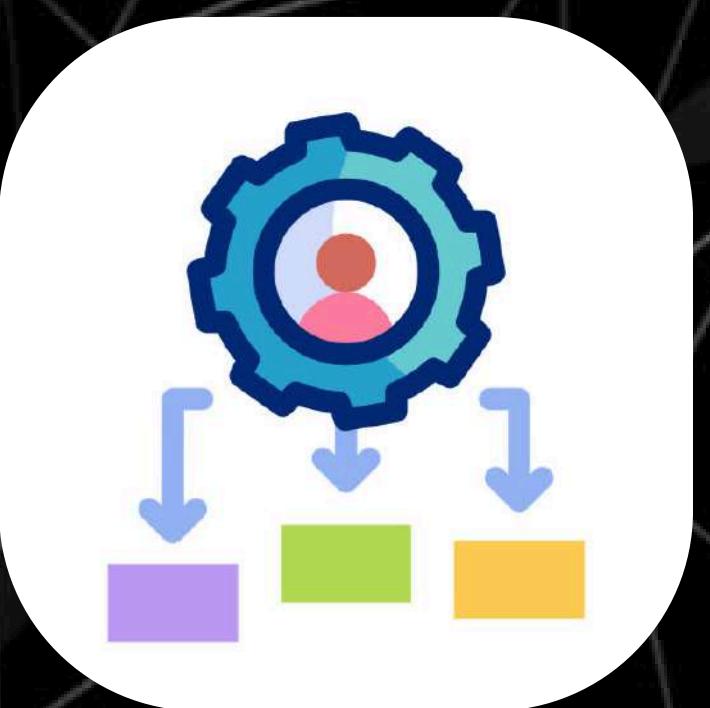
¿COMO LEER ARCHIVOS CSV EN PYTHON?





¿QUÉ ES UN ARCHIVO CSV?

CSV significa Comma Separated Values (valores separados por comas).





EJEMPLO DATOS.csv:

Cada línea es una fila y cada coma separa columnas.



- 1 nombre,edad,nota
- 2 Ana,20,8.5
- 3 Luis,22,7.0
- 4 Carlos,19,9.2



FORMA 1: LEER CSV COMO ARCHIVO DE TEXTO

⚠ Útil solo para entender el formato, no recomendada para trabajo real.



```
1 with open("datos.csv", "r", encoding="utf-8") as archivo:  
2     for linea in archivo:  
3         print(linea.strip())
```



FORMA 2 (RECOMENDADA): USAR EL MÓDULO CSV

Python trae el módulo csv integrado 



LEER CSV FILA POR FILA (CSV.READER)



```
1 import csv
2
3 with open("datos.csv", "r", encoding="utf-8") as archivo:
4     lector = csv.reader(archivo)
5     for fila in lector:
6         print(fila)
```



SALTAR LA FILA DE EN CABEZADOS



```
1 import csv
2
3 with open("datos.csv", "r", encoding="utf-8") as archivo:
4     lector = csv.reader(archivo)
5     next(lector) # salta encabezado
6
7     for fila in lector:
8         print(fila)
```

✓ FORMA 3: LEER CSV COMO DICCIONARIO (CSV.DICTREADER)



Usa los nombres de las columnas.

```
1 import csv  
2  
3 with open("datos.csv", "r", encoding="utf-8") as archivo:  
4     lector = csv.DictReader(archivo)  
5  
6     for fila in lector:  
7         print(fila)
```

ACCEDERA COLUMNAS ESPECÍFICAS



```
1 for fila in lector:  
2     print(fila["nombre"], fila["nota"])
```



VALIDAR SI EL ARCHIVO CSV EXISTE



```
1 import os  
2  
3 if os.path.exists("datos.csv"):  
4     print("Archivo encontrado")
```



BUENAS PRÁCTICAS

- ✓ Usa csv.DictReader para claridad
- ✓ Especifica encoding="utf-8"
- ✓ Valida que el archivo exista
- ✓ No proceses CSV como texto plano en proyectos reales



RESUMEN

- `open()`: Leer como texto
- `csv.reader`: Leer filas como listas
- `csv.DictReader`: Leer filas como diccionarios

**NOS VEMOS EN UN PRÓXIMO
VIDEO DE ESTE CURSO,
SALUDOS**

