SESIÓN 25 · MÓDULO 2

Ficheros, bytes y rutas

Leer y escribir ficheros de texto y binarios con `with`, y moverse por carpetas con `pathlib`.

Antonio Chanes Espigares · CC BY-NC-ND 4.0

Hoy

0:00	Corrección de la tarea 24.
0:15	Situación: un cuaderno: se puede empezar de cero, añadir al final o solo leer.
0:25	**Predice** y visualizador: se ve el disco cambiar.
0:45	Construcción: modos `r`, `w`, `a`, `b`, `with`, `read`, `readline`, `readlines`, `bytes`, `bytearray`,...
1:20	Ejercicios del capítulo.
1:45	Preguntas tipo examen y tarea 25.

Para empezar

Un cuaderno: se puede empezar de cero, añadir al final o solo leer.

Predice

Escribe en papel qué crees que pasará. Después, lo ejecutamos.

```
with open("prueba.txt", "w") as f:
    f.write("uno\ndos\n")
with open("prueba.txt", "a") as f:
    print("tres", file=f)
with open("prueba.txt") as f:
    lineas = f.readlines()
print(len(lineas), lineas[-1].strip())
b = bytearray(b"abc")
b[0] = 65
print(b)
```

Predice

Escribe en papel qué crees que pasará. Después, lo ejecutamos.

```python
with open("prueba.txt", "w") as f:
    f.write("uno\ndos\n")
with open("prueba.txt", "a") as f:
    print("tres", file=f)
with open("prueba.txt") as f:
    lineas = f.readlines()
print(len(lineas), lineas[-1].strip())
b = bytearray(b"abc")
b[0] = 65
print(b)
```

LO QUE SALE

`3 tres` · `bytearray(b'Abc')`

65 es el código de `A`.

Escribir y volver a leer

Ficheros y flujos · 1

```
with open("notas.txt", "w") as f:
    f.write("7\n")
    f.write("9\n")
total = 0
with open("notas.txt") as f:
    for linea in f:
        total = total + int(linea)
print(total)
```

Las formas de leer

Ficheros y flujos · 2

```
with open("lista.txt", "w", encoding="utf-8") as f:
    f.write("pan\nleche\nhuevos\n")

with open("lista.txt", encoding="utf-8") as f:
    print(repr(f.read()))          # todo el texto de golpe

with open("lista.txt", encoding="utf-8") as f:
    print(repr(f.readline()))      # una línea
    print(f.readlines())           # las que quedan, en una lista

with open("lista.txt", encoding="utf-8") as f:
    print([l.strip() for l in f])  # recorriendo: lo más habitual
```

Cuando abrir falla

Ficheros y flujos · 3

```
import errno

try:
    with open("no_existe.txt") as f:
        print(f.read())
except FileNotFoundError as e:
    print("No está:", e.filename)
    print(e.errno == errno.ENOENT, e.strerror)
```

Flujos binarios y `bytearray`

Ficheros y flujos · 4

```
datos = bytearray(4)
for i in range(len(datos)):
    datos[i] = 65 + i
with open("datos.bin", "wb") as f:
    f.write(datos)
print(datos)
```

`pathlib`: las rutas como objetos

Ficheros y flujos · 5

```
print(Path("saludo.txt").is_absolute(), Path("saludo.txt").resolve().is_absolute())
print(sorted(f.name for f in Path(".").iterdir() if f.suffix in (".txt", ".bin")))
```

Manos a la obra

Abre el cuaderno de la sesión en Colab, con los ejemplos de hoy y estos ejercicios:

1. El diario. Escribe un programa que añada la línea "hoy he aprendido…
2. Contar líneas. Escribe un programa que cuente cuántas líneas no…
3. ¿Qué hay en el fichero?
4. A prueba de errores



Cuaderno en Colab

Tarea 25 · Un diario y una copia

1. Con `pathlib`, crea la carpeta `diario` si no existe.
2. Escribe `escribe(fecha, texto)`, que añade el texto al fichero `diario/<fecha>.txt`.
3. Escribe `entradas()`, que devuelve las fechas con entrada, ordenadas (usa `glob` y `stem`).
4. Escribe `lee(fecha)`, que devuelve el texto o `None` si ese día no hay fichero.
5. Escribe `copia(origen, destino)`, que copia un fichero binario en bloques de 1024 bytes y devuelve cuántos bytes copió.

COMPROBACIÓN Dos llamadas a `escribe` con la misma fecha dejan dos líneas en el mismo fichero.

Antes de irte

Las **preguntas tipo examen** del capítulo, en directo: se corrigen al momento.



Preguntas del capítulo

LA PRÓXIMA SESIÓN

Sesión 26 · Cadenas a fondo

Programación en Python · Sesión 25 · Ficheros, bytes y rutas