### Guardar y leer datos generados por el usuario

Guardar datos con json es útil cuando trabajas con datos generados por el usuario, porque si no almacenas la información de tu usuario de alguna manera, la perderás cuando el programa deje de ejecutarse. Veamos un ejemplo en el que pedimos al usuario su nombre la primera vez que ejecuta un programa y luego recordamos su nombre cuando vuelva a ejecutar el programa.

Empecemos por almacenar el nombre del usuario:

**remember\_me.py**

from pathlib import Path  
import json  
  
❶ username = input("What is your name? ")  
  
❷ path = Path('username.json')  
contents = json.dumps(username)  
path.write\_text(contents)  
  
❸ print(f"We'll remember you when you come back, {username}!")

Primero pedimos un nombre de usuario para almacenarlo ❶. A continuación, escribimos los datos que acabamos de recoger en un archivo llamado *username.json* ❷. Luego imprimimos un mensaje informando al usuario de que hemos almacenado su información ❸:

What is your name? Eric  
We'll remember you when you come back, Eric!

Ahora escribamos un nuevo programa que salude a un usuario cuyo nombre ya ha sido almacenado:

**greet\_user.py**

from pathlib import Path  
import json  
  
❶ path = Path('username.json')  
contents = path.read\_text()  
❷ username = json.loads(contents)  
  
print(f"Welcome back, {username}!")

Leemos el contenido del fichero de datos ❶ y luego utilizamos json.loads() para asignar los datos recuperados a la variable username ❷. Como hemos recuperado el nombre de usuario, podemos darle la bienvenida con un saludo personalizado:

Welcome back, Eric!

Necesitamos combinar estos dos programas en un único archivo. Cuando alguien ejecute *remember\_me.py*, queremos recuperar su nombre de usuario de la memoria si es posible; si no, le pediremos un nombre de usuario y lo almacenaremos en *username.json* para la próxima vez. Podríamos escribir aquí un bloque try-except para responder adecuadamente si *username.json* no existe, pero en su lugar utilizaremos un práctico método del módulo pathlib:

**remember\_me.py**

from pathlib import Path  
import json  
  
path = Path('username.json')  
❶ if path.exists():  
 contents = path.read\_text()  
 username = json.loads(contents)  
 print(f"Welcome back, {username}!")  
❷ else:  
 username = input("What is your name? ")  
 contents = json.dumps(username)  
 path.write\_text(contents)  
 print(f"We'll remember you when you come back, {username}!")

Hay muchos métodos útiles que puedes utilizar con los objetos Path. El método exists() devuelve True si existe un archivo o carpeta y False si no existe. Aquí utilizamos path.exists() para averiguar si ya se ha almacenado un nombre de usuario ❶. Si *username.json* existe, cargamos el nombre de usuario e imprimimos un saludo personalizado al usuario.

Si el archivo *username.json* no existe ❷, pedimos un nombre de usuario y almacenamos el valor que introduzca el usuario. También imprimimos el mensaje familiar de que nos acordaremos de ellos cuando vuelvan.

Sea cual sea el bloque que se ejecute, el resultado es un nombre de usuario y un saludo apropiado. Si es la primera vez que se ejecuta el programa, ésta es la salida:

What is your name? Eric  
We'll remember you when you come back, Eric!

Si no:

Welcome back, Eric!

Esta es la salida que ves si el programa ya se ha ejecutado al menos una vez. Aunque los datos de esta sección son sólo una cadena, el programa funcionaría igual de bien con cualquier dato que pueda convertirse a una cadena con formato JSON.

[anterior](c10_33.html)[Subtema 34 de 37: (Ver todo)](c10.html)[siguiente](c10_35.html)