



Clase 3: Guía de Estudio: Manejo de Archivos JSON

1. Introducción a JSON

JSON (JavaScript Object Notation) es un formato ligero para el intercambio de datos, fácil de leer y escribir tanto para humanos como para máquinas. Se utiliza comúnmente en aplicaciones web para el almacenamiento de configuraciones y la transmisión de datos.

Características de JSON:

- **Formato ligero:** Ideal para el intercambio de datos.
- **Estructura legible:** Utiliza llaves `{ }` para objetos y corchetes `[]` para listas.
- **Tipado flexible:** Soporta tipos de datos como cadenas, números, booleanos, arrays y objetos.

2. Lectura de Archivos JSON

Para trabajar con archivos JSON en Python, se utiliza la biblioteca `json`. Aquí tienes un ejemplo detallado de cómo leer un archivo JSON:

Ejemplo de Código:

```
import json

# Abrir el archivo en modo lectura
with open('productos.json', 'r') as archivo:
    productos = json.load(archivo)

# Imprimir cada producto
for producto in productos:
    print(producto)
```

Explicación:

- Se importa la biblioteca `json`.
- Se abre el archivo en modo lectura (`'r'`).
- `json.load()` carga el contenido del archivo en una variable.
- Se itera sobre la lista de productos y se imprimen.

3. Extracción de Datos

Puedes acceder a los datos específicos dentro de los diccionarios. Por ejemplo, para extraer el nombre y el precio de cada producto:

Ejemplo de Código:

```
for producto in productos:
    nombre = producto['nombre']
    precio = producto['precio']
    print(f"Nombre: {nombre}, Precio: {precio}")
```

Explicación:

- Accedemos a las claves específicas dentro del diccionario para obtener el valor correspondiente.

4. Escritura y Adición de Datos 🖋️

Para agregar nuevos datos a un archivo JSON, primero debes cargar el contenido existente, añadir el nuevo dato y luego volver a escribir el archivo.

Ejemplo de Código:

```
# Nuevo producto a añadir
nuevo_producto = {
    "nombre": "Auriculares",
    "precio": 50
}

# Abrir el archivo para leer
with open('productos.json', 'r') as archivo:
    productos = json.load(archivo)

# Añadir nuevo producto
productos.append(nuevo_producto)

# Abrir el archivo para escribir
with open('productos.json', 'w') as archivo:
    json.dump(productos, archivo, indent=4)
```

Explicación:

- Se abre el archivo para leer y se cargan los datos existentes.
- Se añade un nuevo diccionario a la lista.
- Se vuelve a abrir el archivo en modo escritura (`'w'`) y se guarda la lista actualizada.

5. Ejercicios Prácticos 🚀

Ejercicio 1: Leer y mostrar datos de un archivo JSON

1. Crea un archivo `productos.json` con algunos productos (ej. laptops, auriculares).

2. Escribe un script para leer este archivo y mostrar los nombres de los productos.

Solución Paso a Paso:

```
import json

with open('productos.json', 'r') as archivo:
    productos = json.load(archivo)

for producto in productos:
    print(producto['nombre']) # Imprimir nombre de cada producto
```

Ejercicio 2: Añadir un nuevo producto al archivo JSON

1. Define un nuevo producto (ej. un cargador).
2. Modifica el script anterior para añadir este nuevo producto al archivo y guardar los cambios.

Solución Paso a Paso:

```
nuevo_producto = {
    "nombre": "Cargador",
    "precio": 20
}

with open('productos.json', 'r') as archivo:
    productos = json.load(archivo)

productos.append(nuevo_producto)

with open('productos.json', 'w') as archivo:
    json.dump(productos, archivo, indent=4)
```

6. Conclusión y Práctica Continua ✨

El manejo de archivos JSON es una habilidad esencial en el desarrollo de aplicaciones modernas. La práctica constante te permitirá familiarizarte con estos conceptos y aplicarlos efectivamente en tus proyectos.

Ejercicio Adicional:

Convierte un archivo CSV a JSON y viceversa, manteniendo la estructura de los datos. Esto reforzará lo que has aprendido y te proporcionará herramientas útiles para proyectos futuros.

Recuerda, ¡la clave es la práctica! 💪