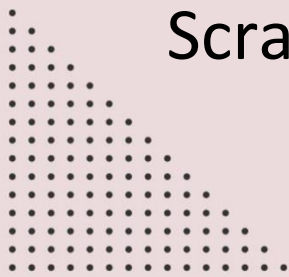


Ejercicio Enunciado



Scraping web fakes



Ejercicio de Web Scraping: FakestoreAPI

Enunciado:

El objetivo de este ejercicio es realizar un web scraping sobre Fake Store API, un sitio web que ofrece una API con datos ficticios de productos, para extraer información de estos productos y guardarla en un archivo CSV.

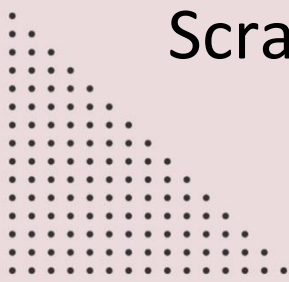
Objetivos específicos:

1. Obtener una lista de productos desde la API.
2. Extraer los siguientes datos para cada producto:
 - a. id
 - b. title
 - c. price
 - d. category
 - e. description
 - f. image
3. Guardar estos datos en un archivo CSV llamado products.csv.

Restricciones:

- Utiliza solo las bibliotecas requests y csv.
- Realiza la solicitud a la API y maneja cualquier error que pueda ocurrir.

Ejercicio Solución



Scraping web fakes



Solución:

1. Instalar Requests: Asegúrate de tener instalada la biblioteca Requests, puedes hacerlo con el siguiente comando:

```
pip install requests
```

2. Escribir el código:

```
import requests
import csv

# URL base de la API de Fakestore
base_url = 'https://fakestoreapi.com/products'

try:
    # Realizar una solicitud GET a la API
    response = requests.get(base_url)
    response.raise_for_status() # Verificar si la solicitud fue exitosa

    # Obtener los datos en formato JSON
    products = response.json()

    # Crear el archivo CSV y escribir los datos
    with open('products.csv', 'w', newline='', encoding='utf-8') as file:
        writer = csv.writer(file)
        # Escribir la cabecera del CSV
        writer.writerow(['ID', 'Title', 'Price', 'Category', 'Description',
            'Image'])
```

```
# Escribir los datos de cada producto
for product in products:
    writer.writerow([
        product['id'],
        product['title'],
        product['price'],
        product['category'],
        product['description'],
        product['image']
    ])
print('Datos guardados en products.csv')
except requests.RequestException as e:
    print(f'Error al realizar la solicitud: {e}')
```

Explicación del código:

- El código realiza una solicitud HTTP GET a la URL de la API para obtener la lista de productos.
- Utiliza el método `response.json()` para convertir la respuesta JSON en una lista de diccionarios de Python.
- Abre el archivo `products.csv` en modo de escritura y usa el módulo `csv` para escribir la cabecera y luego los datos de cada producto.

Este ejercicio ayuda a entender cómo interactuar con una API REST, manejar errores y almacenar datos en un archivo CSV.