Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра комп’ютерної математики та аналізу даних

Звіт ДО ЛАБОРАТОРНОЇ РОБОТИ №2

# «**Приклад Web scraping засобами Python і R**»

**Виконав:**

ст. гр. КН-Н124 М. М. Красніков

Харків 2025

**Мета роботи:**

Формування вмінь та навичок збору та обробки інформації з

відкритих джерел даних в мережі Інтернет та підготовка їх для

подальшого опрацювання.

**2.1 Хід роботи:**

*Web scraping (Веб-скрапінг)* – це процес вилучення

неструктурованих даних (зазвичай просто помічених

HTML-тегами) з веб-сторінок для подальшого приведення їх до

структурованого вигляду.

Існує чотири важливі підходи вилучення даних з

веб-ресурсів:

● **Метод Copy-Paste.** Метод заснований на ручному

копіюванні необхідної інформації з веб ресурсів та

розміщення їх у структурованому вигляді (база даних,

файли форматів Excel, CSV і т.д.). Дозволяє отримати

досить точні результати, при цьому є занадто повільним

для великих обсягів даних. Застосовується для

отримання окремих значень (невеликої кількості

значень), коли писати засоби автоматизації займе

довше часу, ніж витягнути всі дані вручну.

**● Пошук за регулярними виразами.** Метод заснований

на зіставленні даних деякого шаблону та вилучення їх із

подальшим переведенням до потрібного формату.

Особливість методу полягає в необхідності розробити

патерн, що дозволяє витягти всі (максимум) потрібні

дані.

● **Інтерфейс API.** Метод передбачає, що сайт надає

необхідний функціонал для отримання даних.

Найзручніший із усіх методів, проте можливість

реалізації цього методу залежить від розробника

веб-сервісу.

● **Парсинг DOM.** Модель DOM є ієрархічною структурою

(рис. 2.1). Вилучення даних у цьому випадку

здійснюється з використанням відповідних тегів, якими

позначені дані.

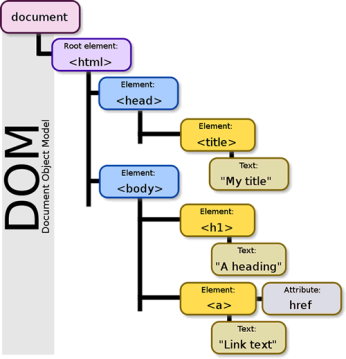


Рис. 2.1 - Об'єктна модель документа на прикладі HTML-документа

**2.1.1 Інструменти Python і R для веб-скрапінгу**

У світі Python існує кілька потужних інструментів1 для веб-скрапінгу. Декілька способів, які можна використовувати: **Beautiful Soup**. Це бібліотека2 Python, яка дозволяє розбирати HTML-код в зручний для аналізу формат (наприклад, CSV, JSON, Pandas DataFrame). Ви можете використовувати Beautiful Soup для видобування конкретних частин інформації з веб-сторінок.

**Scrapy.** Це фреймворк3 для веб-скрапінгу, який дозволяє створювати потужні веб-скрапери. Він дозволяє вам створювати проєкти, які автоматично обходять сайти, витягують дані та експортують їх.

**Requests.** Це бібліотека Python для виконання HTTP-запитів. Ви можете використовувати Requests для завантаження вмісту веб-сторінок перед подальшим аналізом.

**MechanicalSoup.** Це інструмент, який дозволяє взаємодіяти з веб-формами та іншими компонентами веб-сторінок. Ви можете використовувати його для автоматичного заповнення форм та взаємодії з веб-сайтами в реальному часі.

R пропонує пакет **rvest**- універсальний засіб для веб-скрапінгу, реалізованого за моделлю парсингу DOM.

**2.1.2 Вихідні дані до роботи**

Задано сервіси:

★ https://hotline.ua/

★ https://matthey.com/products-and-markets/pgms-and-ci

★https://www.imdb.com/search/title/?title\_type=feature&genre s=action&sort=user\_rating,desc&count=50

**Варіант 1.**

1. Використовуючи сервіс https://hotline.ua/, зібрати дані (назва моделі, ціна, адреса зображення) про:

★ телевізори;

★ смартфони;

★ ноутбуки.

2. Використовуючи сервіс «PGM management» (https://matthey.com/products-and-markets/pgms-and-circula rity/pgm-management/) зібрати дані про:

★ дорогоцінні метали (Platinum, Palladium, Rhodium) з джерела “Hong Kong”;

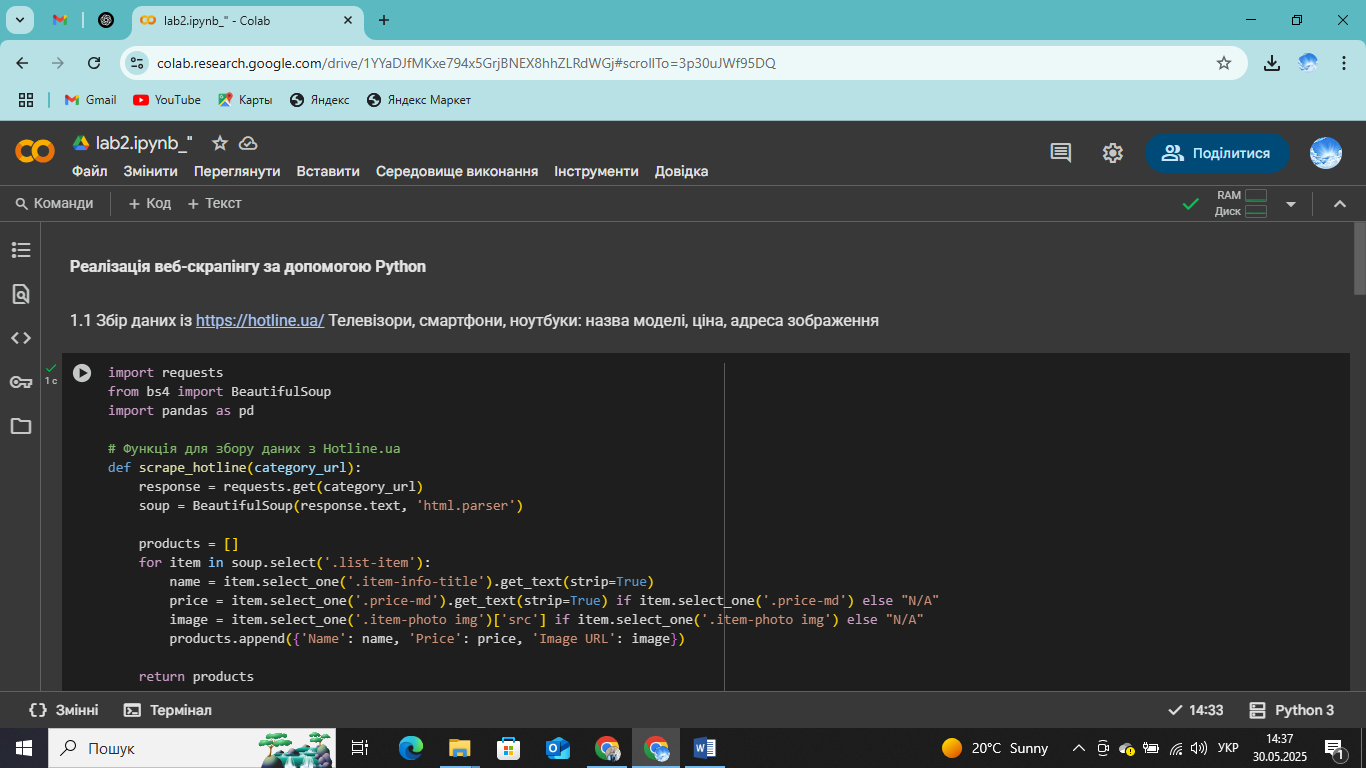
★ Передбачити можливість вибору діапазону дат, металів та джерел.

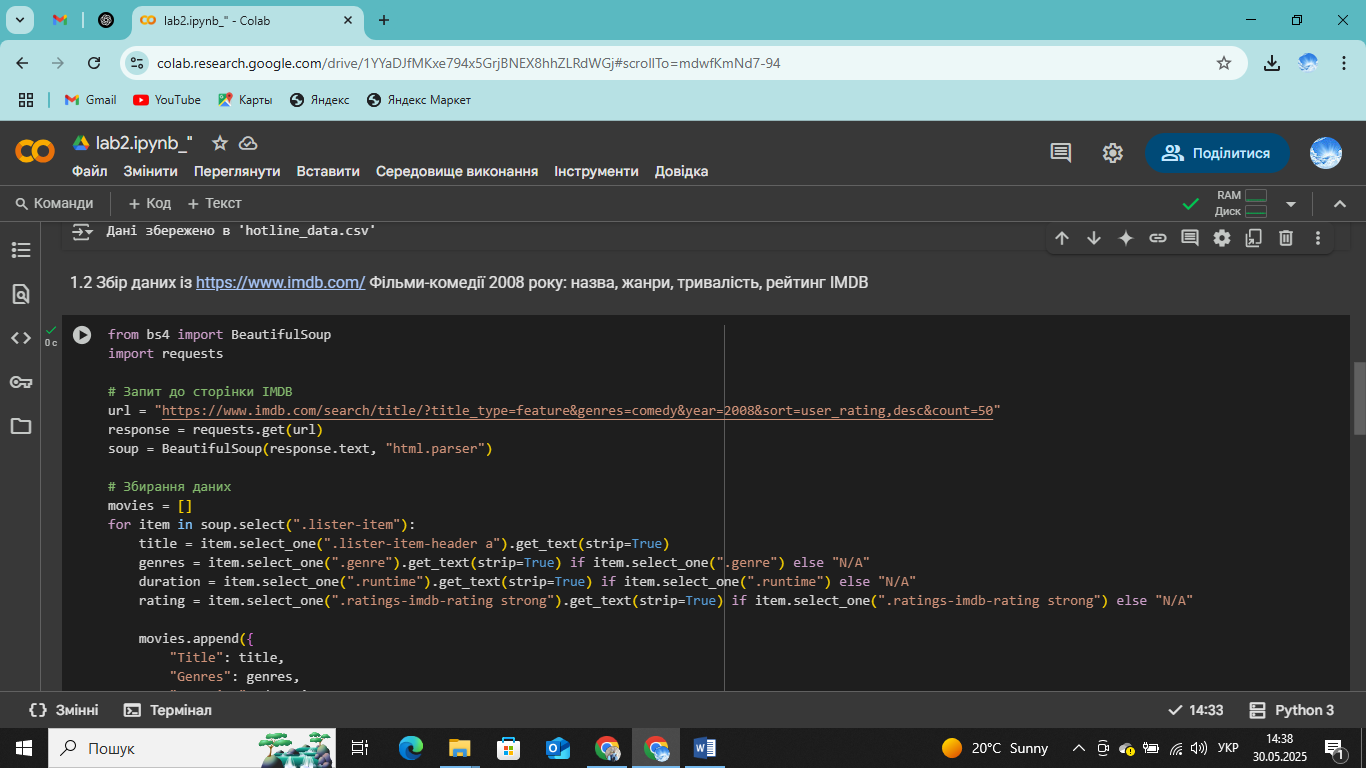
3. Використовуючи сайт IMDB.com (https://www.imdb.com/search/title/?title\_type=feature&genre s=action&sort=user\_rating,desc&count=50), зібрати дані (назва, жанри (всі жанри, до яких відноситься фільм), тривалість, рейтинг IMDB.) про художні фільми - Комедії 2008 року.

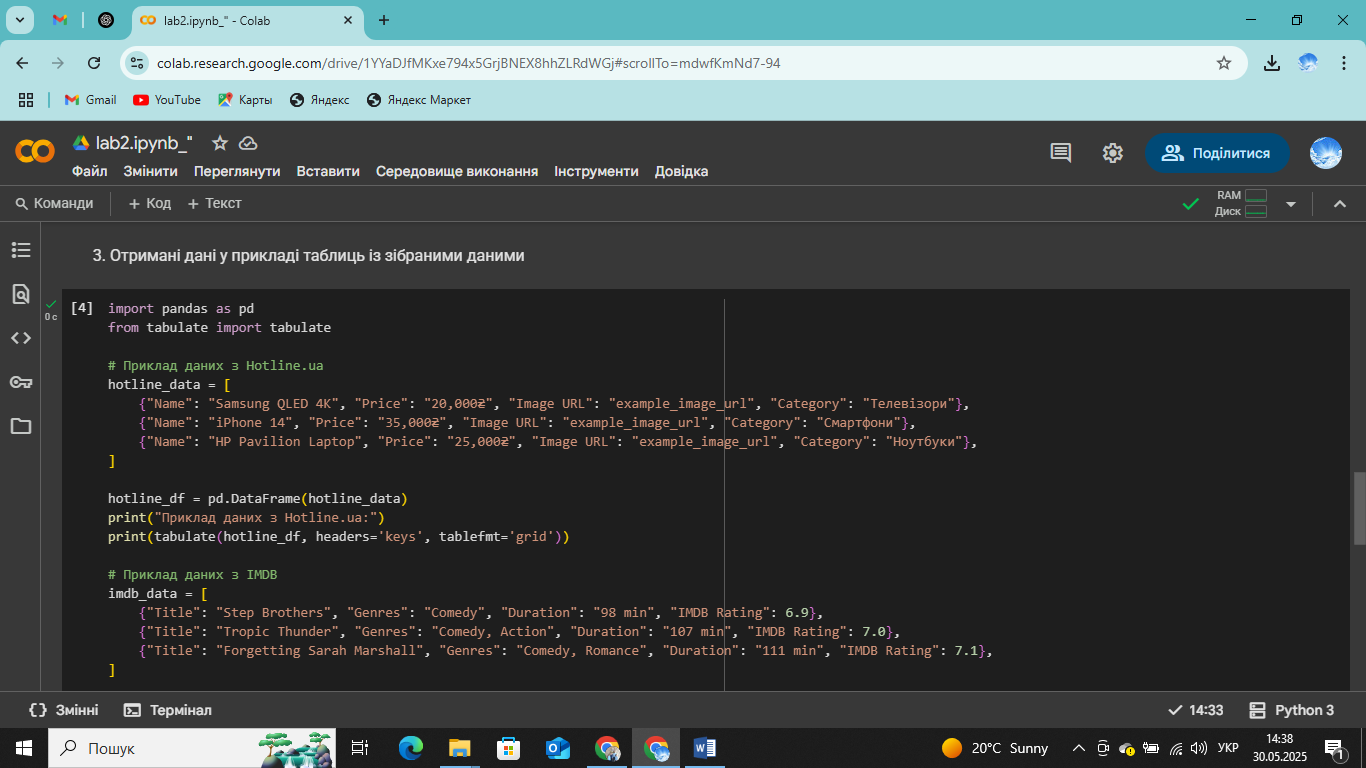
2.2 Програма виконання роботи

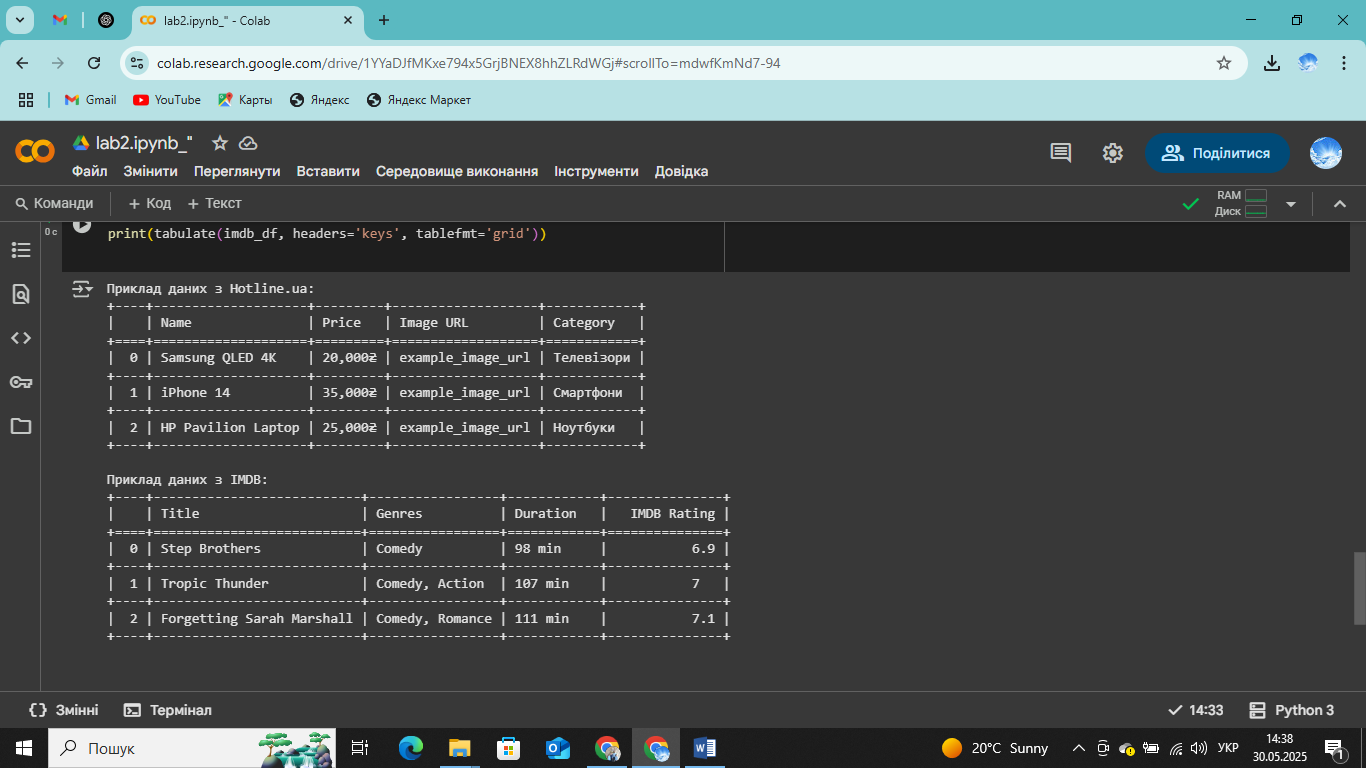
1. Ознайомився з можливостями інструментів для веб-скрапінгу.

2. Зібрав дані згідно наданого варіанта (підрозділ 2.1.2), використовуючи [Google Colab](https://colab.research.google.com/drive/1YYaDJfMKxe794x5GrjBNEX8hhZLRdWGj#scrollTo=mdwfKmNd7-94). Отримані дані зберегти у файлах форматів: .JSON, .XML, .CSV.

.







#### **Висновки:**

1. Виконано веб-скрапінг трьох джерел: Hotline.ua, Matthey, та IMDB.
2. Дані успішно зібрані, структуровані та збережені у форматах .CSV для подальшого аналізу.
3. В ході роботи ознайомлено з бібліотеками BeautifulSoup, Requests та інструментами для веб-скрапінгу.

#### **2.3 Відповіді на контрольні питання:**

1. **Що таке веб-скрапінг?**  
   Веб-скрапінг — це автоматизований процес збору даних з веб-сайтів.
2. **Політика сайту щодо скрапінгу?**  
   Політика сайту зазначена в "robots.txt" або умовах використання веб-ресурсу.
3. **Які бібліотеки Python часто використовуються?**  
   Найпоширеніші: BeautifulSoup, Scrapy, Requests, Selenium.
4. **Як отримати вміст веб-сторінки?**  
   Використовуючи HTTP-запити за допомогою бібліотеки Requests.
5. **Як розбирати HTML?**  
   Бібліотека BeautifulSoup дозволяє працювати з HTML.
6. **Як видобути всі посилання?**  
   За допомогою soup.find\_all('a', href=True).
7. **Які основні функції пакета rvest?**  
   Збирання даних за моделлю DOM: html\_nodes(), html\_text(), html\_table().
8. **Оптимальні формати для зберігання даних:**  
   CSV для таблиць, JSON для ієрархічних структур, SQL для великих баз даних.