**Лабораторна робота №6**

**Тема:** Збір даних з веб-документів за допомогою Python

**Мета:** навчитися одержувати дані з html-сторінок та здійснювати їх аналіз, використовуючи можливості мови Python

**Теоретична частина**

**Збір даних**

Для збору даних з Інтернет-мережі будемо використовувати модуль **requests**, який дозволяє отримувати доступ до веб-сторінок. Як приклад будемо використовувати сайт новин Hacker News.

Згадаймо, що є два найпоширеніші способи доступу до веб-сторінок: запит типу GET і запит типу POST (насправді видів http-запитів набагато більше). Запит типу GET – це коли ви передаєте серверу якусь інформацію в адресному рядку. Наприклад, якщо ви перейдете за такою адресою:

https://translate.google.com.ua/?hl=uk#en/uk/python,

то цим ви просите сервіс Google Translate перевести слово "python" з англійської на українську мову (параметри запиту вказуються після символу "?"). POST-запит – це коли вам потрібно ввести інформацію в яку-небудь форму, наприклад, ввести логін-пароль, який не відображатиметься в адресному рядку браузера.

Ми поки будемо використовувати тільки GET-запити.

Давайте виконаємо два різних GET-запиту до новинного сайту:

>>> import requests

>>> r = requests.get("https://news.ycombinator.com/newest")

>>> r.ok

True

>>> r.status\_code

200

>>> r = requests.get("https://news.ycombinator.com/abrakadabra")

>>> r.ok

False

>>> r.status\_code

404

Перший запит був виконаний успішно, про що говорить значення r.ok і r.status\_code. Другий запит був виконаний до неіснуючої сторінки, що призвело до помилки 404 – "Сторінку не знайдено".

Доступ до вмісту сторінки можна отримати за допомогою атрибута text (для прикладу виведено 100 перших символів):

>>> r.text [: 100]

'<Html op = "newest"> <head> <meta name = "referrer" content = "origin"> <meta name = "viewport" content = "width ='

Як ви бачите це проста HTML-сторінка, з якої нам потрібно витягти цікаву для нас інформацію, а саме:

• заголовок новини;

• автора новини;

• посилання на новину;

• кількість коментарів;

• кількість "лайків", яку набрала стаття.

Наприклад, в наступній новині:

https://dementiy.gitbooks.io/-python/content/assets/Screen%20Shot%202017-01-28%20at%2021.33.29.png

Рисунок 1

• заголовок → Show HN: Pydb – a lightweight database with Python syntax queries, using ZeroMQ;

• автор → asrp;

• посилання → https://github.com;

• кількість коментарів → 11;

• кількість "лайків" → 63.

Для отрмання даних з веб-сторінок є безліч різних модулів. Проблема з HTML в тому, що більшість браузерів поводиться поблажливо, і тому в Інтернеті багато погано-написаних (не по стандарту HTML) HTML-сторінок. Втім, обробка навіть не цілком коректного HTML-коду не така складна, якщо під рукою є відповідні інструменти. Ми будемо користуватися модулем BeautifulSoup 4.

Щоб використовувати BeautifulSoup, потрібно передати функції BeautifulSoup текст веб-сторінки (у вигляді одного рядка). Для уникнення появи попереджень, також слід вказувати назву парсеру (тієї програми, яка здійснює обробку HTML) – з метою сумісності можна використовувати html.parser (він входить в поставку Python і не вимагає установки), але можна також спробувати використовувати html5lib, якщо він встановлений.

>>> from bs4 import BeautifulSoup

>>> page = BeautifulSoup(r.text, 'html.parser')

>>> page

<html op="newest"><head><meta content="origin" name="referrer"><meta content="width=device-width, initial-scal

e=1.0" name="viewport"><link href="news.css?5kjS59ufyw5qyqpjcavc" rel="stylesheet" type="text/css">

<link href="favicon.ico" rel="shortcut icon">

...

Змінна page представляє собою не просто вміст HTML-сторінки, це об'єкт, який дозволяє виконувати запити. Наприклад, ми можемо звернутися до тегу head, а всередині нього до тегу title:

>>> page.head.title

<title>New Links | Hacker News</title>

>>> page.head.title.text

'New Links | Hacker News'

Для того, щоб краще зрозуміти структуру HTML-сторінки слід скористатися веб-інспектором, який є в більшості сучасних браузерів та переглянути код сторінки.

Якщо ви подивіться на структуру HTML-сторінки, то зможете помітити, що є зовнішня таблиця, яка включає в себе ще три таблиці: заголовок, новинну стрічку (яка в свою чергу також складається з великої кількості рядків) і нижній колонтитул (див. рис. 2).

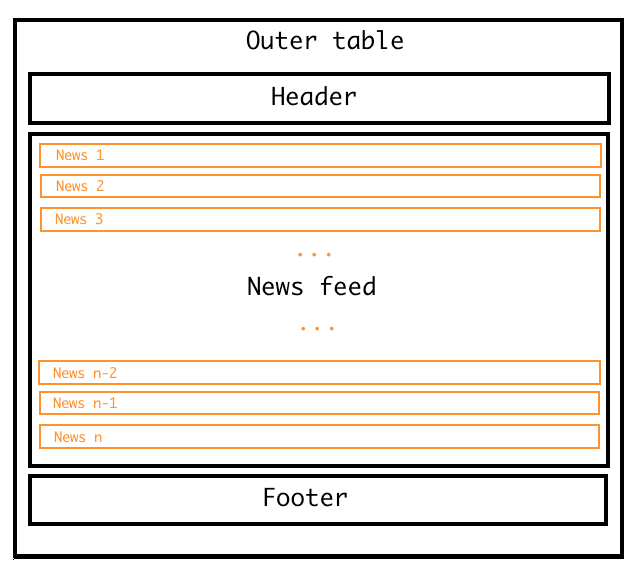


Рисунок 2

Виникає питання як звернутися до внутрішніх таблиць. Якщо ми двічі звернемося до атрибуту table, то отримаємо заголовок:

>>> page.table.table

<table border="0" cellpadding="0" cellspacing="0" style="padding:2px" width="100%"><tr><td style="width:18px;p

adding-right:4px"><a href="http://www.ycombinator.com"><img height="18" src="y18.gif" style="border:1px white

solid;" width="18"/></a></td>

<td style="line-height:12pt; height:10px;"><span class="pagetop"><b class="hnname"><a href="news">Hacker News<

/a></b>

<span class="topsel"><a href="newest">new</a></span> | <a href="newcomments">comments</a> | <a href="show">sho

w</a> | <a href="ask">ask</a> | <a href="jobs">jobs</a> | <a href="submit">submit</a> </span></td><td style="t

ext-align:right;padding-right:4px;"><span class="pagetop">

<a href="login?goto=newest">login</a>

</span></td>

</tr></table>

У об'єкта page крім атрибутів є функції, однією з яких є findAll і дозволяє знайти декілька елементів з однаковими тегами:

>>> tbl\_list = page.table.findAll('table')

>>> len(tbl\_list)

3

Відповідно, нульовий елемент списку tbl\_list це таблиця, яка містить заголовок, перший елемент списку це таблиця з новинами і другий елемент списку це нижній колонтитул

**Збереження даних в sqlite**

В процесі збору даних їх потрібно десь зберігати. Можна, напр., використовувати для зберігання SQLite – компактну вбудовану реляційну базу даних. У стандартній бібліотеці мови Python є модуль sqlite3, який надає інтерфейс для роботи з SQLite. Цей модуль вимагає знання мови SQL, тому ми скористаємося іншою технологією, яка називається ORM.

ORM (англ. Object-relational mapping, рус. Об'єктно-реляційне відображення) – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи "віртуальну об'єктну базу даних". Існують як пропрієтарні, так і вільні реалізації цієї технології.

SQLAlchemy – це бібліотека мовою Python для роботи з реляційними СУБД з застосуванням технології ORM. Служить для синхронізації об'єктів Python і записів реляційної бази даних. SQLAlchemy дозволяє описувати структури баз даних і способи взаємодії з ними на мові Python без використання SQL.

Кожна таблиця описується класом, який повинен успадковуватися від базового класу, створюваного за допомогою функції sqlalchemy.ext.declarative.declarative\_base(). У розглянутому нами прикладі буде тільки один клас – News, з атрибутами: заголовок, автор, посилання, кількість коментарів і число "лайків".

from sqlalchemy.ext.declarative import declarative\_base

Base = declarative\_base()

from sqlalchemy import Column, String, Integer

class News(Base):

\_\_tablename\_\_ = "news"

id = Column(Integer, primary\_key = True)

title = Column(String)

author = Column(String)

url = Column(String)

comments = Column(Integer)

points = Column(Integer)

label = Column(String)

from sqlalchemy import create\_engine

engine = create\_engine("sqlite:///news.db")

Base.metadata.create\_all(bind=engine)

from sqlalchemy.orm import sessionmaker

session = sessionmaker(bind=engine)

s = session()

Нижче наведено приклад створення об'єкту та збереження його в БД:

>>> news = News(title='Lab 7',

author='dementiy',

url='https://dementiy.gitbooks.io/-python/content/lab7.html',

comments=0,

points=0)

>>> news.id, news.title

(None, Lab 7)

>>> s.add(news)

>>> s.commit()

>>> news.id, news.title

(1, Lab 7)

Зверніть увагу, що ідентифікатор об'єкта (id) містить значення None до тих пір, поки ми не зробимо коміт в БД за допомогою методу commit().

Переглянути вміст файлу news.db можна за допомогою програми DB Browser for SQLite.

**Завдання:**

Реалізуйте програму, яка для довільної сторінки будь-якого сайту новин буде підраховувати частоту появи слів у тексті новини, частоту появи html-тегів, кількість посилань та зображень.