**Лабораторная работа №6**

**Тема:** Сбор данных с веб-документов с помощью Python

**Цель:** научиться получать данные с html-страниц и осуществлять их анализ, используя возможности языка Python

**теоретическая часть**

**сбор данных**

Для сбора данных с Интернет-сети будем использовать модуль **requests,** который позволяет получать доступ к веб-страниц. В качестве примера будем использовать сайт новостей Hacker News.

Вспомним, что есть два распространенных способа доступа к веб-страницам: запрос типа GET и запрос типа POST (на самом деле видов http-запросов гораздо больше). Запрос типа GET - это когда вы передаете сервера какую-то информацию в адресной строке. Например, если вы перейдете по адресу:

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

то этим вы просите сервис Google Translate перевести слово "python" с английского на украинский язык (параметры запроса указываются после символа "?"). POST-запрос - это когда вам нужно ввести информацию в какую-нибудь форму, например, ввести логин-пароль, который будет отображаться в адресной строке браузера.

Мы пока будем использовать только GET-запросы.

Давайте выполним два разных GET-запроса к новостному 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

Первый запрос был выполнен успешно, о чем говорит значение r.ok и r.status\_code. Второй запрос был выполнен к несуществующей странице, что привело к ошибке 404 - "Страница не найдена".

Доступ к содержимому страницы можно получить с помощью атрибута text (например выведено 100 первых символов):

>>> r.text [100]

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

Как вы видите это простая HTML-страница, с которой нам нужно извлечь интересующую нас информацию, а именно:

• заголовок новости;

• автора новости;

• ссылки на новость;

• количество комментариев;

• количество "лайков", которую набрала статья.

Например, в следующей новости:

Рисунок 1

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

• автор  asrp;

• ссылки  https://github.com;

• количество комментариев  11;

• количество "лайков"  63.

Для отрмання данных с веб-страниц есть множество различных модулей. Проблема с HTML в том, что большинство браузеров ведет себя снисходительно, и поэтому в Интернете много худо-написанных (не по стандарту HTML) HTML-страниц. Впрочем, обработка даже не вполне корректного HTML-кода не так сложна, если под рукой есть соответствующие инструменты. Мы будем пользоваться модулем BeautifulSoup 4.

Чтобы использовать BeautifulSoup, нужно передать функции BeautifulSoup текст веб-страницы (в виде одной строки). Во избежание появления предупреждений, также следует указывать название парсеру (той программы, которая осуществляет обработку HTML) - с целью совместимости можно использовать html.parser (он входит в поставку Python и не требует установки), но можно попробовать использовать html5lib, если он установлен .

>>> 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">

...

Переменная page представляет собой не просто содержание HTML-страницы, это объект, который позволяет выполнять запросы. Например, мы можем обратиться к тегу head, а внутри него к тегу title:

>>> page.head.title

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

>>> page.head.title.text

"New Links | Hacker News "

Для того, чтобы лучше понять структуру HTML-страницы следует воспользоваться веб-инспектором, который есть в большинстве современных браузеров и посмотреть код страницы.

Если вы посмотрите на структуру HTML-страницы, то сможете заметить, что есть внешняя таблица, которая включает в себя еще три таблицы: заголовок, новостную ленту (которая в свою очередь также состоит из большого количества строк) и нижний колонтитул (см. Рис. 2).

Рисунок 2

Возникает вопрос как обратиться к внутренним таблиц. Если мы дважды обратимся к атрибуту table, то получим 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>

В объекта page кроме атрибутов есть функции, одной из которых является findAll и позволяет найти несколько элементов с одинаковыми тегами:

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

>>> len (tbl\_list)

3

Соответственно, нулевой элемент списка tbl\_list это таблица, которая содержит заголовок, первый элемент списка это таблица с новостями и второй элемент списка это нижний колонтитул

**Сохранение данных в sqlite**

В процессе сбора данных их нужно где-то хранить. Можно, например., Использовать для хранения SQLite - компактную встроенную реляционную базу данных. В стандартной библиотеке языка Python является модуль sqlite3, который предоставляет интерфейс для работы с SQLite.Этот модуль требует знания языка SQL, поэтому мы воспользуемся другой технологии, которая называется ORM.

ORM (англ. Object-relational mapping, рус. Объектно-реляционное отображение) - технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных". Существуют как проприетарные, так и свободные реализации этой технологии.

SQLAlchemy - это библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM. Служит для синхронизации объектов Python и записей реляционной базы данных. SQLAlchemy позволяет описывать структуры баз данных и способы взаимодействия с ними на языке Python без использования SQL.

Каждая таблица описывается классом, который должен наследоваться от базового класса, создаваемого с помощью функции sqlalchemy.ext.declarative.declarative\_base (). В рассматриваемом нами примере будет только один класс - News, с атрибутами: заголовок, автор, ссылки, количество комментариев и число "лайков".

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 ()

Ниже приведен пример создания объекта и сохранение его 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)

Обратите внимание, что идентификатор объекта (id) содержит значение None до тех пор, пока мы не сделаем комит в БД с помощью метода commit ().

Просмотреть содержимое файла news.db можно с помощью программы DB Browser for SQLite.

**Задание:**

Реализуйте программу, которая для произвольной страницы любого сайта новостей будет подсчитывать частоту появления слов в тексте новости, частоту появления html-тегов, количество ссылок и изображений.