**Лабораторна робота №3**

**Парсинг веб-сторінок Python + BeautifulSoup**

*Пилипця Гліба Володимировича*

Звіт стосовно парсингу даних з веб-сторінок на мові програмування Python з використанням бібліотеки BeautifulSoup4, модулів regular expressions, requests.

Beautiful Soup - це парсер для синтаксичного розбору файлів HTML / XML, написаний на мові програмування Python, який може перетворити навіть неправильну розмітку в дерево синтаксичного розбору. Він підтримує прості і природні способи навігації, пошуку та модифікації дерева синтаксичного розбору. У більшості випадків він допоможе програмісту заощадити години і дні роботи. Написаний на мові програмування Ruby порт називається Rubyful Soup.

**Постановка задачі**

Завдання полягатиме в тому, щоб завантажити дані про певні об’єкти: назва , дата і час перегляду, оцінка користувача, ціна, ссилка – прикладів може бути багато; та відобразити ці дані в зручній для користувача формі: база даних, excel файли, текстові документи.

Насправді, можна розбити роботу на 2 етапи:

Етап 1: завантажити і зберегти html-сторінки

Етап 2: розпарсити html в зручний для подальшого аналізу формат (csv, json, pandas dataframe etc.)

**Розбрів поставленої задачі**

Для відправки http-запитів є чимало python-бібліотек, найбільш відомі urllib / urllib2 і Requests. На мій погляд Requests зручніше і лаконічніше, тому використовував її.

Також необхідно вибрати бібліотеку для парсинга html, після перегляду можливих варіантів дійшов до висновку, що найбільш популярними є наступні варіанти:

* Re(<https://docs.python.org/2/library/re.html>)

Регулярні вирази, звичайно річ потрібна , але використовувати тільки їх для парсингу, на мій погляд, не є раціональним шляхом, і вони трохи не для цього. Були придумані більш зручні інструменти для розбору html, так що далі про них.

* [**BeatifulSoup**](http://www.crummy.com/software/BeautifulSoup/bs4/doc/)(<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>) та [**lxml**](http://lxml.de/index.html)(https://lxml.de/index.html)

Це дві найбільш популярні бібліотеки для парсинга html і вибір однієї з них, скоріше, обумовлений особистими уподобаннями. Більш того, ці бібліотеки тісно переплелися: BeautifulSoup став використовувати lxml як внутрішній парсер для прискорення, а в lxml був доданий модуль soupparser. Я парсив дані за допомогою BeautifulSoup і використовуючи XPath селектори в модулі lxml.html.

* [**scrapy**](http://scrapy.org/)(<https://scrapy.org/>)

Це вже не просто бібліотека, а цілий open-source framework для отримання даних з веб-сторінок. У ньому є дуже багато корисних функцій: асинхронні запити, можливість використовувати XPath і CSS селектори для обробки даних, зручна робота з кодуваннями і багато іншого (докладніше можна почитати тут https://doc.scrapy.org/en/latest/intro/overview.html).

1. Причини вибору

В частності я вибрав BeautifulSoup через відстутність великої кількості часу для розбору scrapy, але бажанням навчитися писати парсери. Також, робота з BeautifulSoup представлена великою кількістю відеоматеріалів та документацією(кращою ніж по AngleSharp,C#:).  
Мав таку потребу по роботі зі збору баз даних з лікарів для компанії PremierExpo(для медичної виставки).

1. Установка та початкове ознайомлення

Використання та встановлення бібліотеки було дуже простим та зрозумілим. Як ознайомлення можу порадити відео матеріал <https://www.youtube.com/watch?v=ng2o98k983k>.

Якщо використовувати як середовище розробки Visual Studio, то там більшість бібліотек не викликає труднощів при встановленні(BS4 не є виключенням).

У випадку використання Python IDE або Atom IDE Python необхідно скачати та встановити pip та там ввести відповідну команду для встановлення BS4.  
pip - система управління пакетами, яка використовується для установки і управління програмними пакетами написаними на Python. Багато пакетів можна знайти в Python Package Index. Починаючи з версії Python 2.7.9 і Python 3.4, вони містять пакет pip за замовчуванням.

Загало установка BS4 не є складною, є дуже багато матеріалів стосовно цього в інтернеті.

1. Документація

Документацію практично не використовував, оскільки вона є корисною при детальному розборі певних питань, в той час як для окремих його частин не є зручною(знову ж таки на мій погляд). Для практичних навичок та пошуку конкретних питань використовував stackoverflow, відеоматеріали youtube. Але можу зазначити, що документація доволі обширна.

В частності ось детальна документація на англійській мові <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Частковий переклад з упусканням детальних розборів

<http://wiki.python.su/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8/BeautifulSoup#A.2BBCEEOAQ9BEIEMAQ6BEEEOARHBDUEQQQ6BDgEOQ_.2BBEAEMAQ3BDEEPgRA_HTML>

1. Використання

Поведінка класів та методів є інтуїтивно зрозумілою. Більшість операцій виконується за допомогою команд .find(‘тег’,{словник атрибутів}) та .find\_all(‘тег’,{словник атрибутів})

Для роботи конструктору Beautiful Soup потрібно документ XML або HTML у вигляді рядка (або відкритого файлоподібного об'єкта). Він зробить синтаксичний розбір і створить в пам'яті структуру даних, що відповідає документу.

Під час синтаксичного розбору документа він перекодовується в Unicode. У своїх структурах даних Beautiful Soup зберігає лише рядки Unicode.

Beautiful Soup використовує клас з ім'ям UnicodeDammit для визначення кодування переданих вами документів і перекодування його в Unicode. Якщо це також необхідно для інших документів (без їх синтаксичного розбору за допомогою Beautiful Soup), то можна використовувати UnicodeDammit окремо.

Тобто роботи з різними кодуваннями не складає проблем в BeautifulSoup.

Аналогічно не викликає проблем і навігація та пошук по дереву синтаксичного розбору.

Клас BeautifulSoup містить евристики, повністю аналогічні тим, що застосовуються в веб-браузерах, що дозволяє робити припущення про задуми авторів HTML файлів(тобто парсинг HTML не викликає проблем). Але в XML файлах немає фіксованого порядку тегів такі евристики тут не знадобляться. Тому BeautifulSoup не зможе добре працювати з XML.

Тому можна використовувати клас BeautifulStoneSoup для синтаксичного розбору документів XML. Це основний клас, який не потребує знання діалекту XML.

1. Зауваження

Я використовував BeautifulSoup далеко від максимальних можливостей, тому зауважень надати по даній бібліотеці не можу.

При наявності часу спробував би або детальніше розібрати BeautifulSoup. Якщо точніше, то парсинг даних представлених в складнішому форматі для розпізнавання: дані, з’являються після натискання на кнопки(olx як приклад), парсинг номеру, що міститься на зображенні(avito як приклад).

1. Враження

Написання парсерів на мові програмування Python є значно зручнішим порівняно з C#,C++, тому для початку раджу використовувавти саме її або PHP.

BeautifulSoup проста в початковому розумінні та освоєнні. Про більш складні моменти поки що не можу зазначити.

На першому курсі ЛУН.ua пропонували роботу для першокурсників по написанню парсерів на спеціальному сервісі за допомогою регулярних виразів. Це також було причиною зацікавленості парсерами.

До того ж зазвичай парсинг пов’язаний з Http-запитами, що є корисним для подальшої роботи з веб-сторінками, серверної частини сайту, посилання запитів аутентифікації(це я використав в проекті).

В проекті використовував HTMLAgilityPack як бібліотеку для C# по парсингу.

Для розпізнавання веб-запитів можу порадити програми Fiddler(особливо його) та Inspector веб-браузера.

**Використана література:**

<http://wiki.python.su/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8/BeautifulSoup>

<https://habr.com/post/280238/>

https://doc.scrapy.org/en/latest/intro/overview.html