# Другий етап курсової роботи

**Json** – формат передачі даних; модуль Json у Python може перетворюв ати словник Python у json object, список – у json array, число типу int або float – у json number за допомогою методу dumps та load, зворотні перетворення – через метод loads/JSONDecoder. Велика кількість додаткових параметрів дає змогу зчитувати лише потрібні дані, а також записувати їх у файл у зручному форматі.

1. Зчитування JSON об’єкта. Далі до його складових можна доступатися як до елементів словника

**XML** – стандарт побудови мов розмітки; прикладом xml-мови є HTML. Для роботи з xml можна використовувати модулі lxml, html.parser та інші

Розглянемо вміст пакунку *html.*parser*,* а саме HTMLParser

Парсер, написаний за допомогою цього модуля, може розпізнавати і обробляти початкові теги та їх атрибути, а також вміст, коментарі та кінцеві теги.

Для передачі тексту в парсер використовується метод feed(). Він буде обробляти html-текст, поки не натрапить на останній кінцевий тег, або ж до виклику методу close()(якщо дані згодовуються шматками)

Для обробки переданого тексту використовуються наступні методи: handle\_starttag(для стартових тегів та їх атрибутів) , handle\_data(для вмісту), handle\_startendtag(для одинарних тегів) та handle\_comments(для коментарів)

Наприклад, клас MyHTMLParser додає початкові і кінцеві теги, а також коментарі, у окремі списки. У цьому прикладі на екран буде виведено 3 списки, отримані з html-розмітки сайту The Guardian



Інший модуль для обробки XML – текстів – *xml.* Для ітерацій по елементах( екземпляри класу Element), пошуку за тегами чи атрибутами, побудови власних XML-дерев чи модифікації існуючих зручно використовувати xml.etree.ElementTree. Для швидшої роботи краще імпортувати xml.etree.cElementTree -імплементація на мові C

До елементів дерева можна доступатися за індексом або проходженням по елементах дерева(наприклад, за допомогою методу iter()). Метод find() знаходить перший всі підходящий елемент, findall() – усі підходящі, а iterfind() створює ітератор для перебору їх усіх.

XML підтримує XPath, за допомогою синтаксу якої можна, наприклад шукати у батьківських\дочірніх гілках. Ще можна шукати за атрибутом, і навіть певним його значенням.

Функціональні та нефункціональні вимоги

Функціональні вимоги:

|  |  |
| --- | --- |
| Орієнтовані на процеси | В рамках системи користувач може:   * Голосувати за випадкову книжку і\або запитувати у системи наступну * Додавати власні книжки у окрему “кімнату” * Голосувати лише за книжки з кімнати, до якої можна доступитися за унікальним ключем * Переглядати результати голосування |
| Орієнтовані на інформацію | Система повинна зберігати таку інформацію про книжки:   * З інтернет-магазину або від користувача, що створює кімнату –автор, фотографія, опис, додатково – рейтинг та жанр, * Кількість голосувань за цю книжку у системі |

Нефункціональні вимоги:

|  |  |
| --- | --- |
| Операційні | Система повинна коректно відображатися на будь-якому екрані, у будь-якому браузері |
| Продуктивність | * Час обміну інформацією з сервером не повинен перевищувати 2 секунди * Система повинна працювати 24\7, 365 днів на рік |
| Безпека | * Лише користувачі, що ввійшли через Twitter або Facebook, зможуть користуватися системою |
| Культурні та політичні вимоги | * Можна додавати інформацію про книги будь-якою мовою. Мова веб-сайту - англійська |

Дані, отримані з Інтернету

Для кожної книги з її сторінки у інтернет-магазині будуть отримані такі дані:

* Автор: рядок(стрінг)
* Назва книжки: рядок
* Зображення обкладинки: посилання на зображення на сайті
* Жанр: рядок
* Опис: рядок(текст)