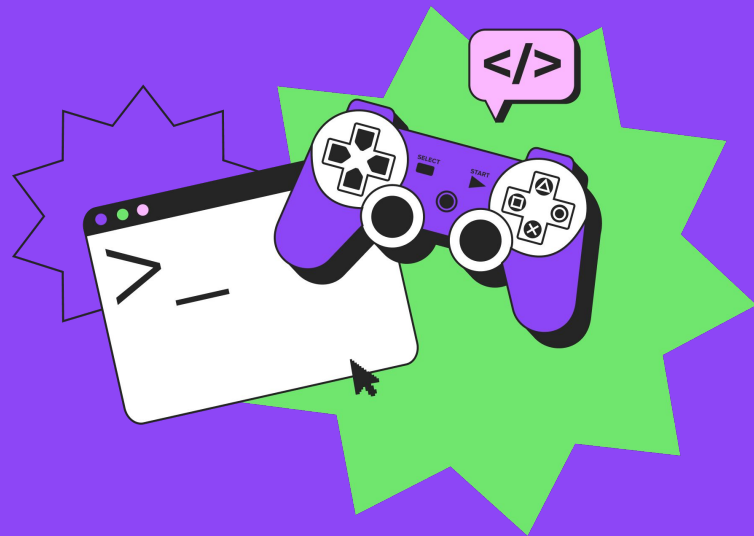


Парсинг HTML. XPath

Семинар 4
Сбор и разметка данных





Сбор и разметка данных

1

Основы клиент-серверного взаимодействия. Парсинг API.

2

Парсинг HTML. BeautifulSoup.

3

СУБД MongoDB и ClickHouse в Python

4

Парсинг HTML. XPath.

5

Scrapy.

6

Scrapy. Парсинг фото и файлов.

7

Selenium в Python.




8

Работа с данными.

9

Инструменты разметки наборов данных.

Что будет на уроке сегодня

-  Как использовать библиотеку lxml для парсинга содержимого HTML.
-  Различные типы выражений XPath и их использование для выбора определенных элементов и атрибутов.
-  Как извлекать данные из определенных элементов и атрибутов с помощью lxml и XPath.





Викторина



Какова цель XPath при парсинге HTML и скрейпинге?

1. Для отправки HTTP-запросов на веб-сайты.
2. Для извлечения данных с веб-страниц.
3. Для имитации веб-браузера.
4. Для организации доступа к частям документа XML



Какова цель XPath при парсинге HTML и скрейпинге?

1. Для отправки HTTP-запросов на веб-сайты.
2. Для извлечения данных с веб-страниц.
3. Для имитации веб-браузера.
4. Для организации доступа к частям документа XML



Что является примером оси XPath?

1. ancestor
2. sibling
3. descendant
4. все вышеперечисленное



Что является примером оси XPath?

1. ancestor
2. sibling
3. descendant
4. все вышеперечисленное



Что из нижеперечисленного НЕ является преимуществом использования селекторов CSS над выражениями XPath для веб-скрейпинга?

1. Более лаконичный синтаксис
2. Более гибкий и мощный
3. Лучшая производительность
4. Легче понять и прочитать



Что из нижеперечисленного НЕ является преимуществом использования селекторов CSS над выражениями XPath для веб-скрейпинга?

1. Более лаконичный синтаксис
2. Более гибкий и мощный
3. Лучшая производительность
4. Легче понять и прочесть



Какое выражение XPath выберет все элементы <a>, включая атрибут href, содержащие слово "example"?

1. `//a[contains(@href, 'example')]`
2. `//a[contains(@attribute, 'example')]`
3. `//a[contains(@value, 'example')]`
4. `//a[contains(@class, 'example')]`



Какое выражение XPath выберет все элементы <a>, включая атрибут href, содержащие слово "example"?

1. `//a[contains(@href, 'example')]`
2. `//a[contains(@attribute, 'example')]`
3. `//a[contains(@value, 'example')]`
4. `//a[contains(@class, 'example')]`



Каково назначение функции `lxml.html.fromstring()` при парсинге HTML?

1. Для отправки HTTP-запросов на веб-сайты
2. Для преобразования содержимого HTML в объект XML
3. Для парсинга содержимого HTML в древовидную структуру
4. Для кодирования содержимого HTML в определенный формат



Каково назначение функции `lxml.html.fromstring()` при парсинге HTML?

1. Для отправки HTTP-запросов на веб-сайты
2. Для преобразования содержимого HTML в объект XML
3. Для парсинга содержимого HTML в древовидную структуру
4. Для кодирования содержимого HTML в определенный формат



Что из перечисленного ниже является потенциальной этической проблемой при веб-скрейпинге?

1. Использование выражений XPath для извлечения данных
2. Доступ к данным в Интернете, защищенным авторским правом
3. Сохранение данных в CSV-файл
4. Скрейпинг данных без разрешения



Что из перечисленного ниже является потенциальной этической проблемой при веб-скрейпинге?

1. Использование выражений XPath для извлечения данных
2. Доступ к данным в Интернете, защищенным авторским правом
3. Сохранение данных в CSV-файл
4. **Скрейпинг данных без разрешения**



Что из перечисленного ниже является ограничением использования выражений XPath для веб-скрейпинга?

1. Сложность в изучении и использовании
2. Низкая производительность по сравнению с другими методами
3. Выражения XPath могут потребовать обновления при изменении HTML-структуры веб-страницы
4. Ограниченная поддержка регулярных выражений



Что из перечисленного ниже является ограничением использования выражений XPath для веб-скрейпинга?

1. Сложность в изучении и использовании
2. Низкая производительность по сравнению с другими методами
3. Выражения XPath могут потребовать обновления при изменении HTML-структуры веб-страницы
4. Ограниченная поддержка регулярных выражений



Вопросы?

Вопросы?



Вопросы?





Практика



Знакомство с целевым веб-сайтом

<https://worldathletics.org/records/toplists/sprints/60-metres/indoor/women/senior/2023?page=1>



Задание 1

Напишите сценарий на языке Python, который выполняет следующие задачи:

- отправляет HTTP GET-запрос на целевой URL и получает содержимое веб-страницы.
- выполняет парсинг HTML-содержимого ответа с помощью библиотеки lxml.
- используя выражения XPath, извлеките данные из первой строки таблицы.
- выведите извлеченные данные из первой строки таблицы в консоль.



20 минут

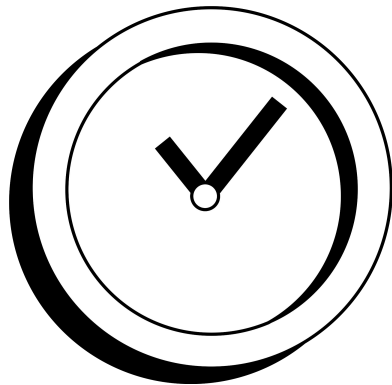


Задание 1

Напишите сценарий на языке Python, который выполняет следующие задачи:

- отправляет HTTP GET-запрос на целевой URL и получает содержимое веб-страницы.
- выполняет парсинг HTML-содержимого ответа с помощью библиотеки lxml.
- используя выражения XPath, извлеките данные из первой строки таблицы.
- выведите извлеченные данные из первой строки таблицы в консоль.

<<20:00-





Задание 2

Модифицируйте предыдущий сценарий, чтобы получить данные из таблицы.

Требования:

- Создайте пустой список, который будет хранить словари с данными из каждой строки
- Извлеките соответствующие данные из каждой строки таблицы и сохраните их в словаре.
- Добавьте каждый словарь к списку данных.
- Выведите полученный список в консоль.



30 минут



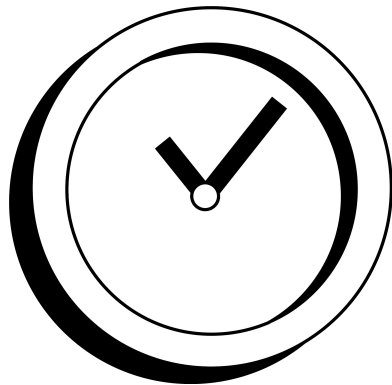
Задание 2

Модифицируйте предыдущий сценарий, чтобы получить данные из таблицы.

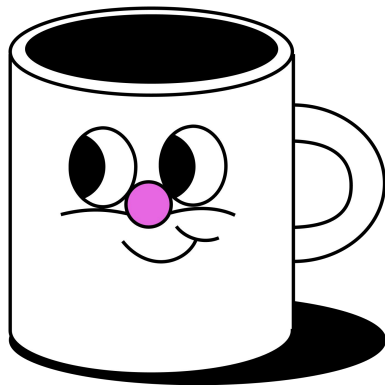
Требования:

- Создайте пустой список, который будет хранить словари с данными из каждой строки
- Извлеките соответствующие данные из каждой строки таблицы и сохраните их в словаре.
- Добавьте каждый словарь к списку данных.
- Выведите полученный список в консоль.

<<20:00-



Перерыв



<<5:00->>



Задание 3

- Определите базовый URL для страниц записей, включая параметр номера страницы.
- Создайте функцию для скрейпинга данных таблицы из одной страницы.
- Создайте функцию для сохранения полученных данных в базе данных MongoDB.
- Создайте главную функцию, которая выполняет итерации по всем страницам записей и сохраняет данные для каждой страницы.
- Ваш скрипт Python должен быть модульным, с отдельными функциями для скрейпинга и сохранения данных, чтобы его можно было легко модифицировать и расширять. Он также должен включать задержку между запросами не менее 5 секунд, чтобы не перегружать сервер.



40 минут



Задание 3 - Hints

- Используйте функцию `time.sleep()` для добавления задержки между запросами. Это важно для того, чтобы не перегружать сервер и не быть заблокированным.
- Используйте User Agent, чтобы избежать блокировки сервером.
- Протестируйте свой код на небольшом количестве страниц, прежде чем запускать его на всем наборе. Это поможет вам выявить ошибки и отладить функции скрейпинга и сохранения данных.



40 минут



Домашнее задание

1. Выберите веб-сайт с табличными данными, который вас интересует.
2. Напишите код Python, использующий библиотеку requests для отправки HTTP GET-запроса на сайт и получения HTML-содержимого страницы.
3. Выполните парсинг содержимого HTML с помощью библиотеки lxml, чтобы извлечь данные из таблицы.
4. Сохраните извлеченные данные в CSV-файл с помощью модуля csv.

Ваш код должен включать следующее:

1. Строку агента пользователя в заголовке HTTP-запроса, чтобы имитировать веб-браузер и избежать блокировки сервером.
2. Выражения XPath для выбора элементов данных таблицы и извлечения их содержимого.
3. Обработка ошибок для случаев, когда данные не имеют ожидаемого формата.
4. Комментарии для объяснения цели и логики кода.

Примечание: Пожалуйста, не забывайте соблюдать этические и юридические нормы при веб-скреппинге.



Вопросы?

Вопросы?



Вопросы?





Спасибо за внимание!