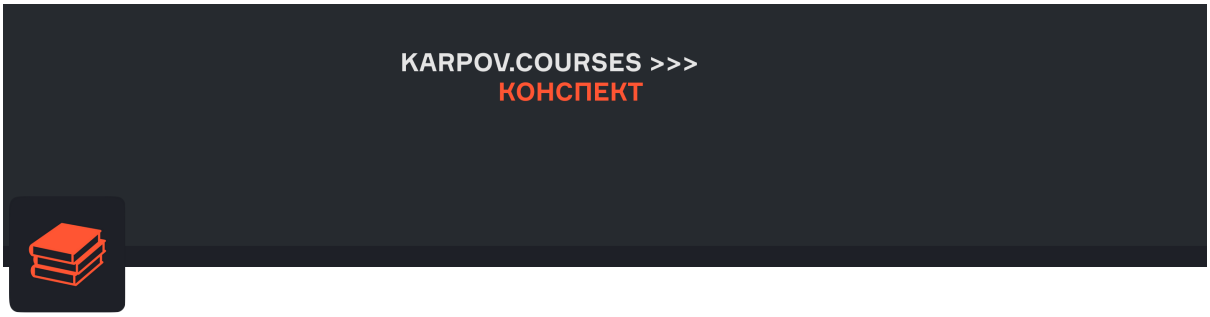


> Конспект > 9 урок > Парсинг данных из интернета, сбор датасетов

> Оглавление

> Использование парсинга

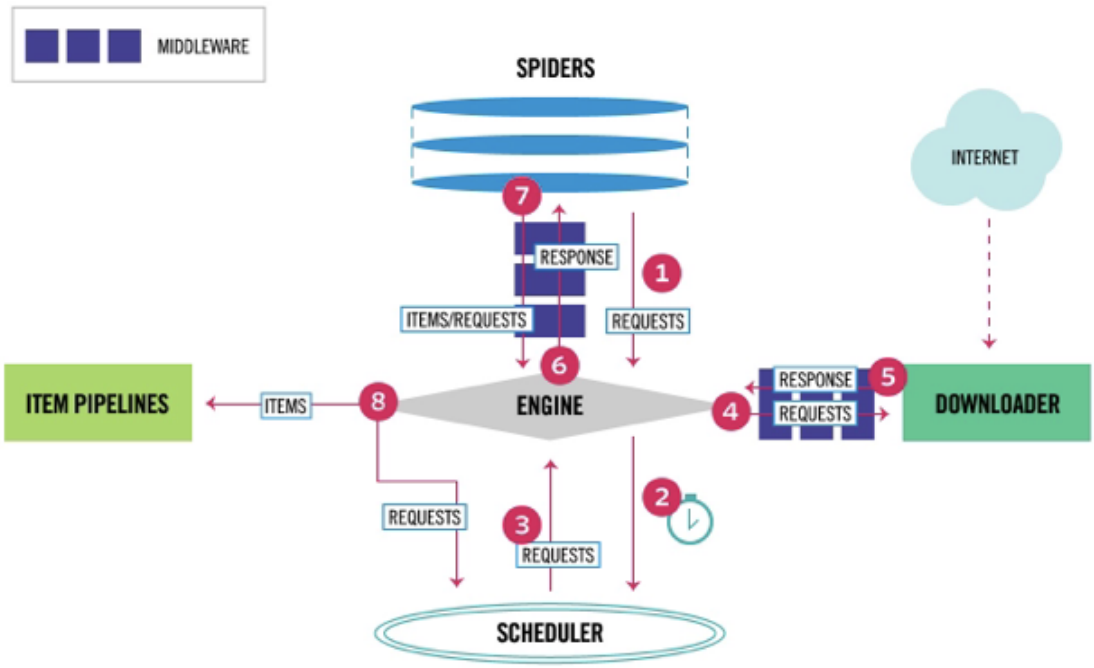
Парсинг — это автоматизированный процесс сбора данных с сайтов, применяемый для получения контента и дополнительной информации в достаточных объёмах.

- Парсинг зачастую является первым этапом в воронке матчинга;
- С точки зрения законности находится в "серой зоне" — всё зависит от целей сбора данных и их дальнейшего использования;

- Парсинг полезен для создания бейзлайнов и pet-projects;
- Некоторые рабочие задачи неочевидно решаются парсингом.

> Подходы

- Проверить, нужен ли парсинг: можно обойтись API-запросами или найти готовый датасет;
- Простые http(s)-запросы (requests) и обработка их ответов — вся обработка ложится на программиста;
- Библиотека `BeautifulSoup` — отвечает за удобный парсинг структуры сайта, делегирует вам весь процесс вокруг этого;
- Библиотека `Scrapy` — отвечает за полный пайплайн работы с веб-данными, от мультипоточной обкачки страниц до дедупликации спаршенных объектов в БД.



Пример пайплайна со Scrapy

> Advanced топики

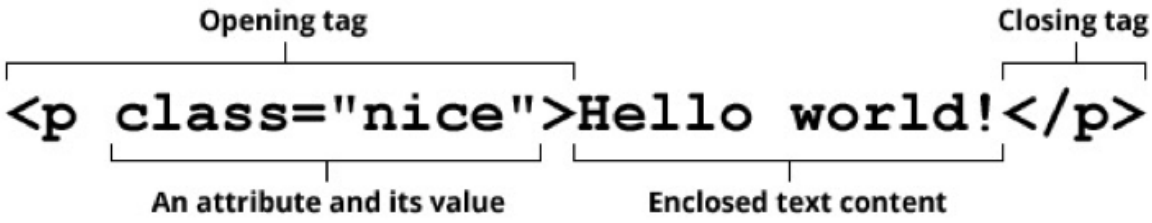
Что делать, чтобы вас не забанили:

- `Selenium` — библиотека для непосредственного управления браузером, имитации кликов и действий пользователя;
- Cookies — хранилище мета-информации о пользователе (пустые = подозрительно);
- Proxy — для подмены IP-адресов в запросах;
- User Agent — для имитации запроса конкретного устройства/браузера.

> Quickstart в HTML

HTML (HyperText Markup Language — "язык гипертекстовой разметки") — самый базовый строительный блок веб-сайтов.

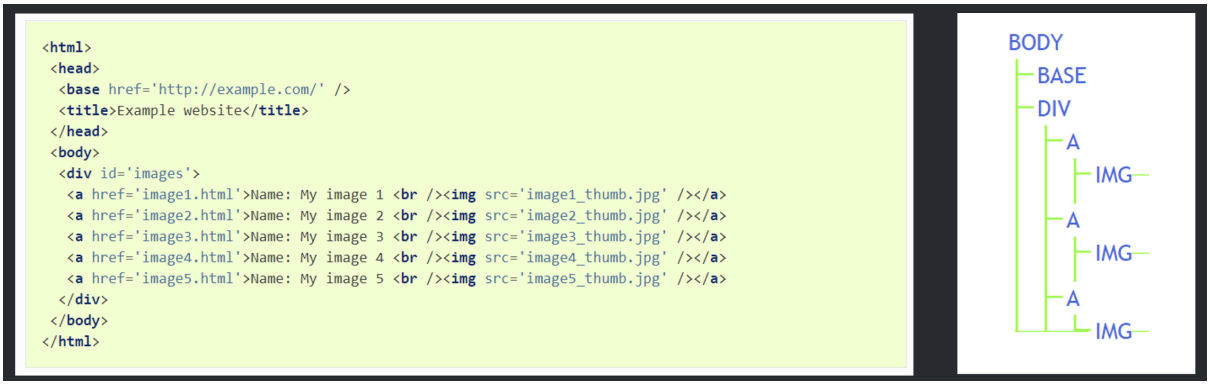
- Документ HTML — это простой текст, разделённый на элементы;
- Элементы окружены одинаковыми открывающимися и закрывающимися тегами, каждый тег начинается и заканчивается с угловых скобок (<>);
- Есть теги, которые созданы не для добавления текста, например `<img>`
- Тегам могут соответствовать атрибуты, параметризующие свойства блока, ограниченного тегом.



Пример структуры с тегом p и атрибутом class со значением "nice"

Структуру `HTML` страницы можно представить в виде дерева.

В левой части изображён пример кода, справа — его визуализация с подуровнями вложения:



Пример HTML-кода и соответствующего ему дерева

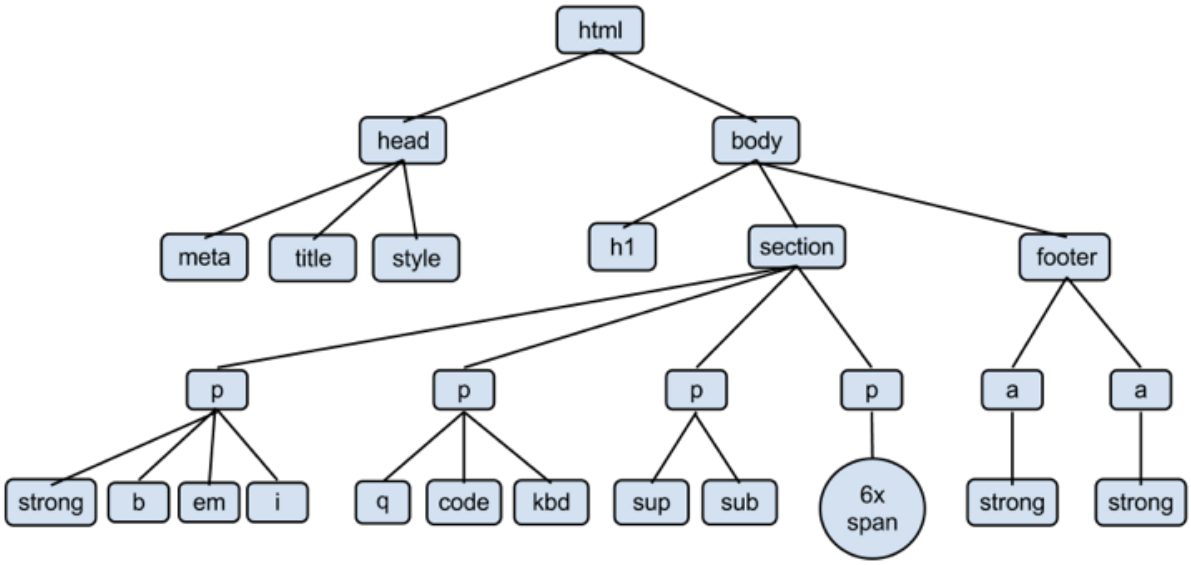
Видно, что существует некоторый верхний уровень с названием `body`, в нём содержится блок `<div>`. Этот элемент является блочным и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Большая часть работы в парсинге связана с ним, так как, по стандартам интернет-вёрстки, хорошей практикой является оперирование именно тегом `<div>`.

В данном случае в этот контейнер помещено несколько ссылок на изображения, и потому его атрибут `id` логично назван images.

Внутри блока каждый тег `<a>` формирует кликабельную ссылку, в атрибуте которой указана страница для перехода после нажатия, а внутрь ссылки вставлено изображение с уже озвученным тегом `<img>`, и это изображение вложено в ссылку. На практике это означает, что при клике и на текст, и на картинку произойдёт переход. Однако тут важнее вложенная структура, и по дереву справа видно, что `<img>` как бы внутри элемента `<a>`.

Поскольку документ можно представить в виде дерева, а значит и графа, то к нему применимы понятия родителя, листовой ноды, узла дерева, потомка и т.д.

У ноды могут быть атрибуты, примеры которых мы уже рассмотрели. Они помогают отличить её от других нод.



Пример визуализации сложной страницы

В примере видны разные паттерны, визуально выделяется большой блок `body`, в котором и расположена большая часть контента. Ниже, в элементе `section`, есть несколько абзацев, каждый из которых имеет своих потомков.

Правила для дерева:

- Есть корневой элемент;
- У элемента дерева есть предки и могут существовать потомки;
- Элемент находится на определённом уровне вложенности;
- У элементов определён порядок: возможна индексация элементов.

Навигация по дереву может происходить при помощи `XPath`.

> XPath

XPath — язык запросов к элементам `XML` или `XHTML` документа, представленным в виде древовидной структуры.

Строка `XPath` — это фактически путь к элементу в дереве, где каждый уровень разделяется косой чертой "/".

> Пример

```
<html>
 <body>
    <div id="first">
      <span>span внутри div</span>
    </div>
    <div id="second"></div>
    <span>некоторый текст</span>
  </body>
</html>
```

На выделенный объект указывает следующий XPath: /html/body/*/span

---

`<span>` — это тег-обёртка для строковых блоков, никакой дополнительной нагрузки без указания атрибутов он не несёт и изменений в визуализацию веб-страницы не вносит. Такую же функцию выполняет тег `<div>`, только он служит контейнером не для строк, а для других блоков.

В примере:

1. Начинаем с корня, на что указывает первая наклонная черта;
2. Переходим сначала в блок `html`, хранящий в себе весь код страницы;
3. Потом в элемент `body`, тем самым отсекая блоки с метаинформацией и прочим мусором, который часто бывает на сайтах;
4. Затем, как и в регулярных выражениях, ставим звёздочку для поиска среди любого узла с любым названием — так, чтобы внутри был элемент `span`

> Оси

Для более сложных конструкций для выборки относительно текущего узла используются специальные конструкции, которые называются оси. Их порядка 20 штук.

Базовые оси:

- `self::` ("`.`") — текущий элемент;

- `parent::` ("`..`") — родительский узел;
- `attribute::` ("`@`") — работа с атрибутами;
- `//` — полное множество потомков (не следующий уровень, а именно все элементы в поддереве, которые имеют в качестве родительской ноды текущий или указанный элемент);
- `child::` (можно опустить как самый частый) — прямые потомки.

> Пример с добавлением осей

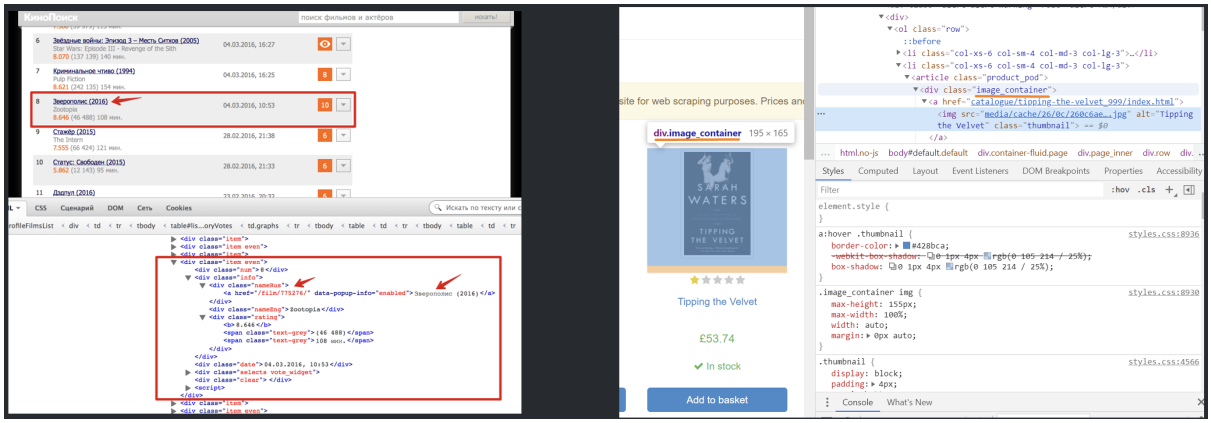
Запишем аналогичные `/html/body/*/span` варианты обращения с применением осей:

- `/child::html/child::body/child::*/child::span` — с прописыванием всех указателей на элементы.
- `.//*[@id="first"]/span` — поиск по всем объектам дерева, где атрибут `id` равен строке first. Выражение в квадратных скобках называется предикатом — фильтрация.
- `.//*[text()="span внутри div"]` — поиск объекта, текст которого равен строке "span внутри div".
- `//*[@id="second"]/../div[1]/span[1]` — если дана информация, что необходимый узел находится в первом блоке `div` на том же уровне, что и блок с `@id="second"`, конструкция избыточна в столь примитивном случае, однако может использоваться в комплексных ситуациях.

> Способ получения XPath

Можно использовать разные способы, в том числе терминал Scrapy или просмотр кода страницы прямо в браузере:

1. Ctrl + Shift + i — переход в консоль браузера (альтернативный вариант: "Просмотреть код" в контекстном меню страницы).
2. Elements — нужная вкладка.


КиноПоиск
поиск фильмов и актёров
Звездные войны: Эпизод 3 – Месть Ситхов (2005)
Криминальное чтиво (1994)
Зверополис (2016)
Стажёр (2015)
Статус: Свободен (2015)
Дрянь (2016)
CSS
Сценарий
DOM
Сеть
Cookies
site for web scraping purposes. Prices and
div.image_container 195 × 165
Tipping the Velvet
£53.74
In stock
Add to basket
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Console
What's New