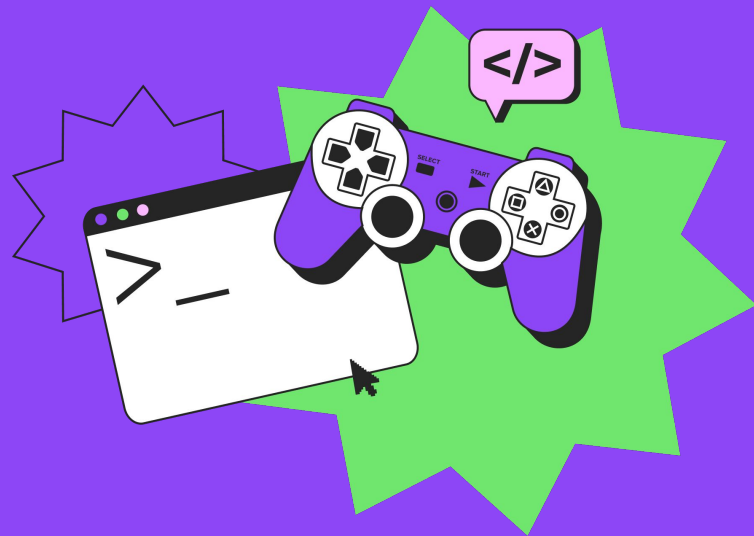






# Парсинг HTML. Beautiful soup

Семинар 2

Сбор и разметка данных



## Что будет на уроке сегодня

-  Будем создавать скрипты Python для запроса веб-страниц и парсинга содержимого HTML с помощью BeautifulSoup.
-  Извлекать данные из определенных HTML-тегов и атрибутов с помощью BeautifulSoup.
-  Определять структуру и иерархию HTML-контента для эффективного парсинга.
-  Применять BeautifulSoup на различных веб-сайтах для извлечения нужных данных.

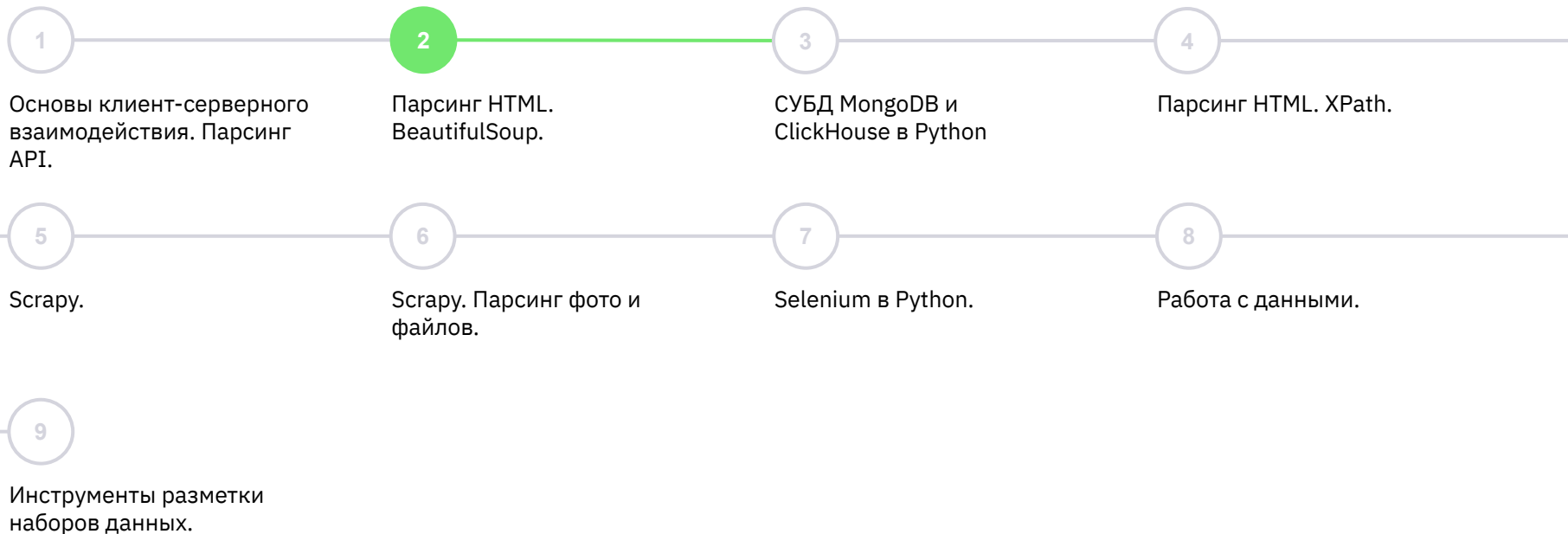




# Викторина



## Сбор и разметка данных





**Что из нижеперечисленного не является распространенным форматом данных, используемым при веб-скреппинге?**

1. JSON
2. CSV
3. HTML
4. XML



**Что из нижеперечисленного не является распространенным форматом данных, используемым при веб-скреппинге?**

1. JSON
2. CSV
3. HTML
4. XML



## **Что Какая из следующих техник используется для того, чтобы избежать блокировки при веб-скреппинге?**

1. Рандомизация строк пользовательского агента
2. Ограничение количества запросов в минуту
3. Ротация IP-адресов
4. Все вышеперечисленное



## Что Какая из следующих техник используется для того, чтобы избежать блокировки при веб-скреппинге?

1. Рандомизация строк пользовательского агента
2. Ограничение количества запросов в минуту
3. Ротация IP-адресов
4. Все вышеперечисленное





## Что из перечисленного ниже не является шагом в веб-скрейпинге?

1. Отправка HTTP-запросов на веб-сайт
2. Парсинг содержимого веб-сайта
3. Хранение данных в базе данных
4. Анализ данных для получения информации



## Что из перечисленного ниже не является шагом в веб-скрейпинге?

1. Отправка HTTP-запросов на веб-сайт
2. Парсинг содержимого веб-сайта
3. Хранение данных в базе данных
4. Анализ данных для получения информации



## Каков правильный синтаксис для создания объекта BeautifulSoup?

1. `soup = BeautifulSoup('http://example.com', 'html.parser')`
2. `soup = BeautifulSoup('http://example.com')`
3. `soup = BeautifulSoup('<html><body><h1>Example</h1></body></html>')`
4. `soup = BeautifulSoup('<html><body><h1>Example</h1></body></html>', 'html.parser')`



## Каков правильный синтаксис для создания объекта BeautifulSoup?

1. `soup = BeautifulSoup('http://example.com', 'html.parser')`
2. `soup = BeautifulSoup('http://example.com')`
3. `soup = BeautifulSoup('<html><body><h1>Example</h1></body></html>')`
4. `soup = BeautifulSoup('<html><body><h1>Example</h1></body></html>', 'html.parser')`



**Какой метод в BeautifulSoup используется для поиска первого вхождения тега?**

1. `find`
2. `find_all`
3. `select`
4. `get`



**Какой метод в BeautifulSoup используется для поиска первого вхождения тега?**

1. `find`
2. `find_all`
3. `select`
4. `get`



## Каково назначение регулярных выражений в Python?

1. Для работы со строками
2. Для выполнения веб-скрепинга
3. Для форматирования текста
4. Для поиска закономерностей в тексте



## Каково назначение регулярных выражений в Python?

1. Для работы со строками
2. Для выполнения веб-скрепинга
3. Для форматирования текста
4. Для поиска закономерностей в тексте





## Каково назначение заголовка "User-Agent" в HTTP-запросе?

1. Для аутентификации пользователя
2. Для указания типа отправляемых данных
3. Для указания кодировки содержимого
4. Для определения типа браузера или клиента, выполняющего запрос.



## Каково назначение заголовка "User-Agent" в HTTP-запросе?

1. Для аутентификации пользователя
2. Для указания типа отправляемых данных
3. Для указания кодировки содержимого
4. Для определения типа браузера или клиента, выполняющего запрос.



## Какой метод в BeautifulSoup используется для поиска всех вхождений тега?

1. find
2. find\_all
3. select
4. get



Вопросы?

Вопросы?



Вопросы?





# Практика



## Знакомство с целевым веб-сайтом

[https://www.boxofficemojo.com/intl/?ref=bo\\_nb\\_hm\\_tab](https://www.boxofficemojo.com/intl/?ref=bo_nb_hm_tab)



## Задание 1

- установите библиотеку BeautifulSoup.
- создайте новый сценарий Python и импортируйте библиотеку BeautifulSoup.
- напишите код для запроса веб-страницы [https://www.boxofficemojo.com/intl/?ref=bo\\_nb\\_hm\\_tab](https://www.boxofficemojo.com/intl/?ref=bo_nb_hm_tab) с помощью библиотеки requests.
- выведите HTML-содержимое веб-страницы в консоль.



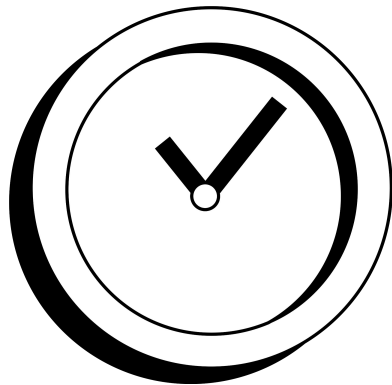
10 минут



## Задание 1

- установите библиотеку BeautifulSoup.
- создайте новый сценарий Python и импортируйте библиотеку BeautifulSoup.
- напишите код для запроса веб-страницы [https://www.boxofficemojo.com/intl/?ref=bo\\_nb\\_hm\\_tab](https://www.boxofficemojo.com/intl/?ref=bo_nb_hm_tab) с помощью библиотеки requests.
- выведите HTML-содержимое веб-страницы в консоль.

<<10:00-







## Задание 2

Напишите сценарий на языке Python, чтобы получить ссылки на релизы фильмов со страницы "International Box Office" на сайте Box Office Mojo.

Сохраните ссылки в списке и выведите список на консоль.

Требования:

- Используйте библиотеку requests для запроса веб-страницы.
- Используйте BeautifulSoup для парсинга HTML-содержимого веб-страницы.
- Найдите все ссылки в колонке #1 Release веб-страницы.
- Используйте библиотеку urllib.parse для объединения ссылок с базовым URL.
- Сохраните ссылки в списке и выведите список на консоль.



20 минут



## Задание 2

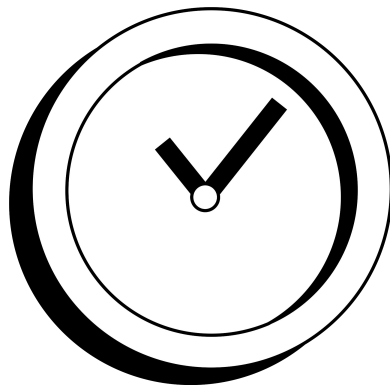
Напишите сценарий на языке Python, чтобы получить ссылки на релизы фильмов со страницы "International Box Office" на сайте Box Office Mojo.

Сохраните ссылки в списке и выведите список на консоль.

Требования:

- Используйте библиотеку requests для запроса веб-страницы.
- Используйте BeautifulSoup для парсинга HTML-содержимого веб-страницы.
- Найдите все ссылки в колонке #1 Release веб-страницы.
- Используйте библиотеку urllib.parse для объединения ссылок с базовым URL.
- Сохраните ссылки в списке и выведите список на консоль.

<<20:00-





## Задание 3

- в консоли разработчика браузера Chrome найдите таблицу с данными и изучите ее HTML-структуру
- напишите сценарий Python для запроса веб-страницы и парсинга HTML-содержимого с помощью библиотеки BeautifulSoup.
- извлеките данные из таблицы и сохраните их в списке словарей, где каждый словарь представляет строку данных.
- преобразуйте список словарей в pandas DataFrame и выведите его на консоль.



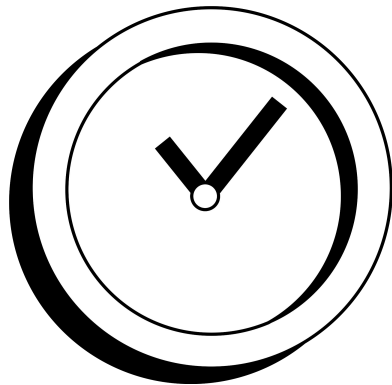
**40 минут**



## Задание 3

- в консоли разработчика браузера Chrome найдите таблицу с данными и изучите ее HTML-структуру
- напишите сценарий Python для запроса веб-страницы и парсинга HTML-содержимого с помощью библиотеки BeautifulSoup.
- извлеките данные из таблицы и сохраните их в списке словарей, где каждый словарь представляет строку данных.
- преобразуйте список словарей в pandas DataFrame и выведите его на консоль.

<<40:00-

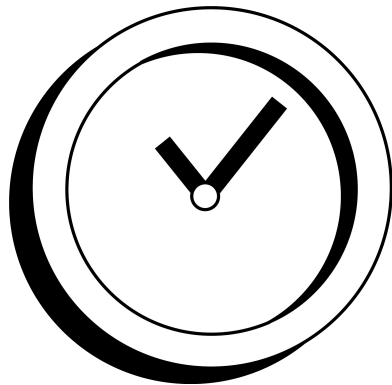




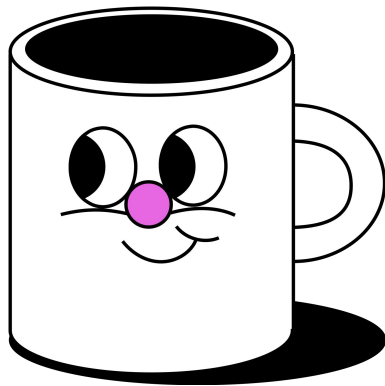
## Задание 3 - Hints

- Чтобы извлечь данные из таблицы и сохранить их в списке словарей, вы можете перебирать каждую строку таблицы с помощью цикла for.
- Для каждой строки создайте новый словарь для хранения данных.
- Используйте список заголовков, чтобы определить ключ для каждого значения в строке.
- Используйте метод `find_all()`, чтобы найти все ячейки в строке, и с помощью атрибута `text` извлеките текстовое содержимое каждой ячейки.
- Возможно, вам понадобится преобразовать некоторые данные в другой тип данных, например, преобразовать сумму в долларах в целое число.
- Наконец, присоедините словарь к списку (тип `list`).

<<40:00-



## Перерыв



<<5:00->>



## Задание 4

Ваша задача - создать Python-сценарий, который извлекает данные по каждому фильму ( по каждой ссылке, сохраненной в url\_joined) и сохраняет их в JSON-файл.

- Для каждого фильма извлеките следующую информацию:

Distributor

Opening (в формате int)

Release Date

MPAA

Running Time (в секундах)

Genres (в виде списка)

In Release

Widest Release (в формате int)

- Сохраните извлеченные данные в виде списка словарей.

- Добавьте временную задержку в 10 секунд между каждым запросом, чтобы не перегружать сервер.

- сохраните извлеченные данные в JSON-файл с именем 'box\_office\_data.json'.



40 минут



## Задание 4

Ваша задача - создать Python-сценарий, который извлекает данные по каждому фильму ( по каждой ссылке, сохраненной в url\_joined) и сохраняет их в JSON-файл.

- Для каждого фильма извлеките следующую информацию:

Distributor

Opening (в формате int)

Release Date

MPAA

Running Time (в секундах)

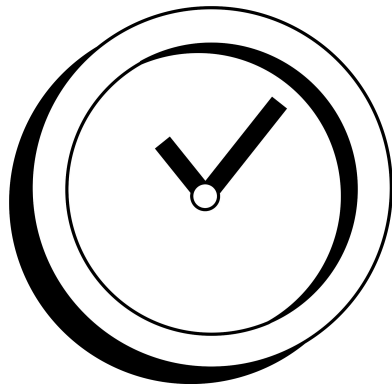
Genres (в виде списка)

In Release

Widest Release (в формате int)

- Сохраните извлеченные данные в виде списка словарей.
- Добавьте временную задержку в 10 секунд между каждым запросом, чтобы не перегружать сервер.
- сохраните извлеченные данные в JSON-файл с именем 'box\_office\_data.json'.

<<40:00-







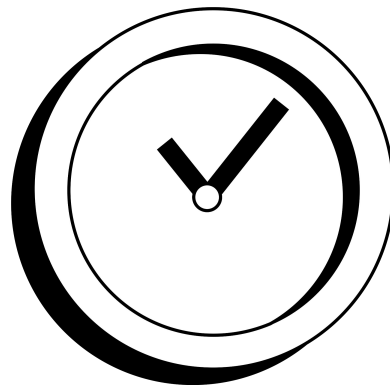
## Задание 4 - Hints

- начните с импорта необходимых библиотек в начале вашего сценария:

```
import requests
from bs4 import BeautifulSoup
import urllib.parse
from datetime import datetime, time, timedelta
import time
import re
import json
```

- пройдите по каждому URL в url\_joined, и для каждого URL
- установите в заголовке User-Agent правильное значение.

<<40:00-

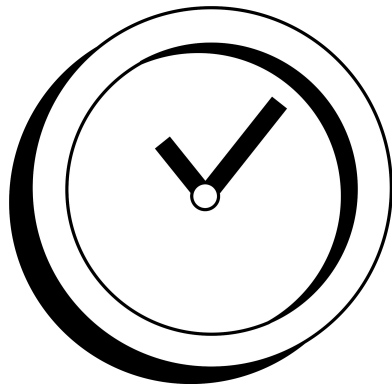




## Задание 4 - Hints

- пройдитеесь в цикле по каждому URL в `url_joined`, и для каждого URL:
- используйте объект `soup` чтобы извлечь таблицу, содержащую информацию о фильмах. Это можно сделать, найдя элемент `<div>` с классом `'a-section a-spacing-none mojo-summary-values mojo-hidden-from-mobile'`.
- извлеките данные из каждой строки таблицы и сохраните в словаре. Это можно сделать, найдя все элементы `<div>` с классом `'a-section a-spacing-none'` внутри таблицы, и затем извлечь текстовое содержимое первого тега `<span>` и второго тега `<span>` внутри каждого из этих элементов.
- Используйте функцию `time.sleep(10)`, чтобы сделать паузы после каждой итерации цикла.

<<40:00-





## Домашнее задание

Выполнить скрейпинг данных в веб-сайта <http://books.toscrape.com/> и извлечь информацию о всех книгах на сайте во всех категориях: название, цену, количество товара в наличии (In stock (19 available)) в формате integer, описание.

Затем сохранить эту информацию в JSON-файле.



Вопросы?

Вопросы?



Вопросы?





Спасибо за внимание!