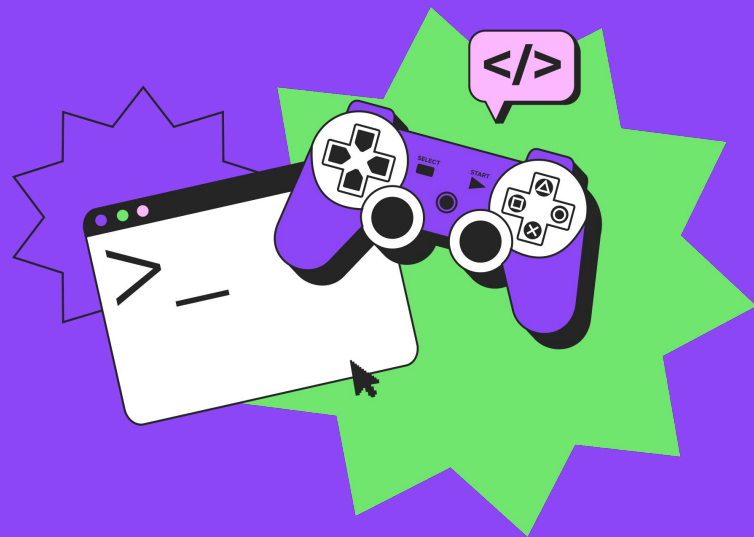


# Основы клиент-серверного взаимодействия. Парсинг API

Семинар 1

Сбор и разметка данных

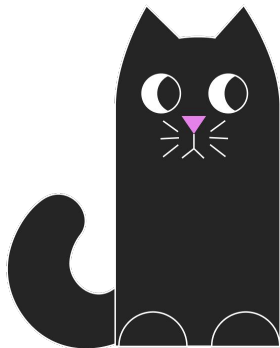




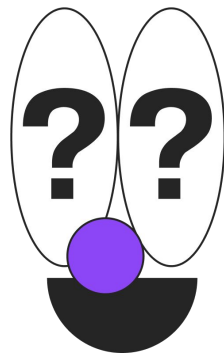
# Знакомство и содержание урока



**Ответьте на несколько вопросов  
сообщением в чат**



**Из какого вы города?**



**Сколько вам лет?**



**Кем вы работаете сейчас?  
Как долго?**



# 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

-  Цикл "запрос-ответ" и его работа.
-  Различные типы взаимодействия клиента и сервера.
-  Различные типы API, включая REST и SOAP.
-  Важность анализа HTTP-запросов и ответов.
-  Основы форматов ответов JSON и XML.





# Викторина



## Что представляет собой цикл "запрос-ответ" при взаимодействии клиент-сервер?

1. Процесс, в котором клиент запрашивает данные у сервера, а сервер отвечает.
2. Процесс, в котором сервер запрашивает данные у клиента, а клиент отвечает на запрос
3. Процесс, в котором клиент обрабатывает данные на сервере
4. Процесс, в котором сервер обрабатывает данные на клиенте



## Что представляет собой цикл "запрос-ответ" при взаимодействии клиент-сервер?

1. Процесс, в котором клиент запрашивает данные у сервера, а сервер отвечает.
2. Процесс, в котором сервер запрашивает данные у клиента, а клиент отвечает на запрос
3. Процесс, в котором клиент обрабатывает данные на сервере
4. Процесс, в котором сервер обрабатывает данные на клиенте





## Что такое REST API?

1. Тип API, который использует модель взаимодействия `stateful`
2. Тип API, использующий модель взаимодействия `stateless`
3. Тип API, использующий формат данных XML
4. Тип API, использующий формат данных JSON



## Что такое REST API?

1. Тип API, который использует модель взаимодействия stateful
2. Тип API, использующий модель взаимодействия stateless
3. Тип API, использующий формат данных XML
4. Тип API, использующий формат данных JSON



## **Какой уровень отсутствует в модели Open System Interconnection (OSI)?**

1. Физический (Physical)
2. Канальный (Data Link)
3. Сеть (Network)
4. Транспорт (Transport)
5. Структура приложений (Application Framework)



## Какой уровень отсутствует в модели Open System Interconnection (OSI)?

1. Физический (Physical)
2. Канальный (Data Link)
3. Сеть (Network)
4. Транспорт (Transport)
5. Структура приложений (Application Framework)



## В чем разница между JSON и XML?

1. JSON - это текстовый формат данных, а XML - двоичный формат данных
2. JSON - это формат данных, используемый для обмена данными, а XML - это язык разметки, используемый для представления данных
3. JSON - это формат данных, используемый для представления данных, а XML - это формат данных, используемый для обмена данными
4. JSON - это язык разметки, используемый для представления данных, а XML - это текстовый формат данных, используемый для обмена данными



## В чем разница между JSON и XML?

1. JSON - это текстовый формат данных, а XML - двоичный формат данных
2. JSON - это формат данных, используемый для обмена данными, а XML - это язык разметки, используемый для представления данных
3. JSON - это формат данных, используемый для представления данных, а XML - это формат данных, используемый для обмена данными
4. JSON - это язык разметки, используемый для представления данных, а XML - это текстовый формат данных, используемый для обмена данными



## Каков синтаксис для доступа к коду состояния ответа HTTP в Python?

1. `response.status`
2. `response.status_code`
3. `response.code`
4. `response.http_code`



## Каков синтаксис для доступа к коду состояния ответа HTTP в Python?

1. `response.status`
2. `response.status_code`
3. `response.code`
4. `response.http_code`





## Каково назначение аргумента `params` в функции `requests.get()` в Python?

1. Для указания данных, которые должны быть отправлены в теле запроса
2. Для указания параметров запроса к API
3. Чтобы указать `headers` для запроса API
4. Чтобы указать учетные данные аутентификации для API



## Каково назначение аргумента `params` в функции `requests.get()` в Python?

1. Для указания данных, которые должны быть отправлены в теле запроса
2. Для указания параметров запроса к API
3. Чтобы указать `headers` для запроса API
4. Чтобы указать учетные данные аутентификации для API



Вопросы?

Вопросы?



Вопросы?





# Практика



## Настройка среды разработки

Необходимо установить Visual Studio Code (VS Code) и дистрибутив Anaconda, если они еще не установлены.

Создать виртуальную среду в Anaconda, где установить необходимые для семинара пакеты: requests, json, pandas, matplotlib, pymongo.

Создать новый блокнот в Google Colab.



## Задание 1

- использовать библиотеку requests в Python для отправки запросов GET, POST, PUT и DELETE на конечную точку REST API <https://jsonplaceholder.typicode.com/posts/1>.
- использовать методы requests.get(), requests.post(), requests.put() и requests.delete() для отправки соответствующих HTTP-запросов.
- проверить код состояния ответа и вывести текст ответа, если запрос был успешным.

### Инструкции:

можно использовать предоставленный код в качестве отправной точки для своего решения.



**30 минут**



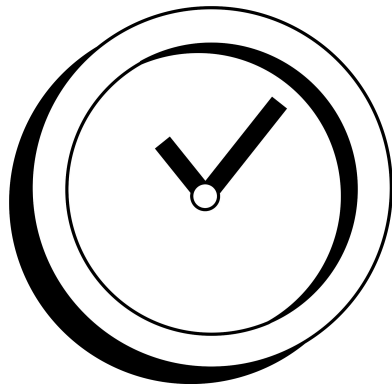
## Задание 1

- использовать библиотеку requests в Python для отправки запросов GET, POST, PUT и DELETE на конечную точку REST API <https://jsonplaceholder.typicode.com/posts/1>.
- использовать методы `requests.get()`, `requests.post()`, `requests.put()` и `requests.delete()` для отправки соответствующих HTTP-запросов.
- проверить код состояния ответа и вывести текст ответа, если запрос был успешным.

### Инструкции:

можно использовать предоставленный код в качестве отправной точки для своего решения.

<<30:00-





## Задание 2

- использовать библиотеку requests в Python для отправки запросов GET, POST, PUT и DELETE на конечную точку REST API

<https://jsonplaceholder.typicode.com/posts/1>.

- использовать методы requests.get(), requests.post(), requests.put() и requests.delete() для отправки соответствующих HTTP-запросов.

- проверить код состояния ответа и вывести текст ответа, если запрос был успешным.

Инструкции:

Можно использовать предоставленный код в качестве отправной точки для своего решения.



30 минут





## Задание 2

- использовать библиотеку requests в Python для отправки запросов GET, POST, PUT и DELETE на конечную точку REST API

<https://jsonplaceholder.typicode.com/posts/1>.

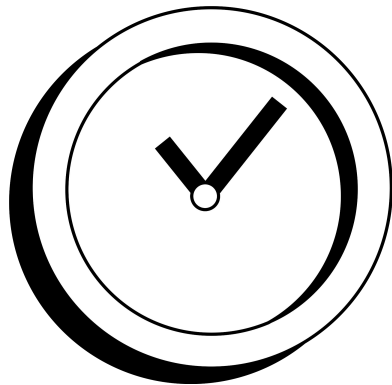
- использовать методы `requests.get()`, `requests.post()`, `requests.put()` и `requests.delete()` для отправки соответствующих HTTP-запросов.

- проверить код состояния ответа и вывести текст ответа, если запрос был успешным.

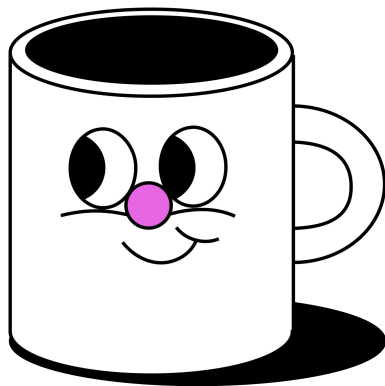
Инструкции:

Можно использовать предоставленный код в качестве отправной точки для своего решения.

<<30:00-



## Перерыв



<<5:00->>



## Задание 3

- Создайте сценарий Python, который запрашивает данные из Foursquare API с помощью библиотеки requests.
- Сценарий должен предложить пользователю ввести название города.
- Затем сценарий должен отправить запрос в Foursquare API для поиска ресторанов в указанном городе.
- Сценарий должен обработать ответ API и извлечь название и адрес каждого ресторана.
- Скрипт должен вывести название и адрес каждого ресторана в консоль.

### Требования:

Использовать API Foursquare для получения данных.

Использовать библиотеку requests для отправки запросов API

Использовать библиотеку json для обработки ответа API.

Запросить у пользователя название города

Извлечь и вывести название и адрес каждого ресторана из ответа API.



40 минут

## Задание 3

- Создайте сценарий Python, который запрашивает данные из Foursquare API с помощью библиотеки requests.
- Сценарий должен предложить пользователю ввести название города.
- Затем сценарий должен отправить запрос в Foursquare API для поиска ресторанов в указанном городе.
- Сценарий должен обработать ответ API и извлечь название и адрес каждого ресторана.
- Скрипт должен вывести название и адрес каждого ресторана в консоль.

### Требования:

Использовать API Foursquare для получения данных.

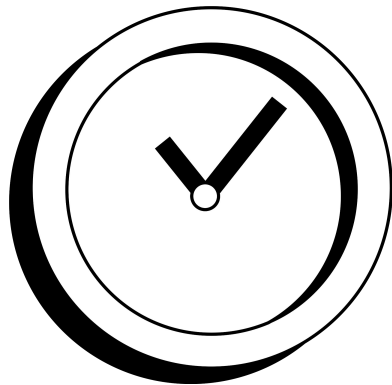
Использовать библиотеку requests для отправки запросов API

Использовать библиотеку json для обработки ответа API.

Запросить у пользователя название города

Извлечь и вывести название и адрес каждого ресторана из ответа API.

<<40:00-





## Задание 4

Работа с API для сбора и обработки данных, а затем создание DataFrame для подготовки данных к анализу.

- Перенести код из блока 3 в Google Colab.
- Модифицировать код, чтобы извлечь адрес, название ресторана, а также координаты (долгота и широта)
- Создать pandas DataFrame из полученных данных



**20 минут**

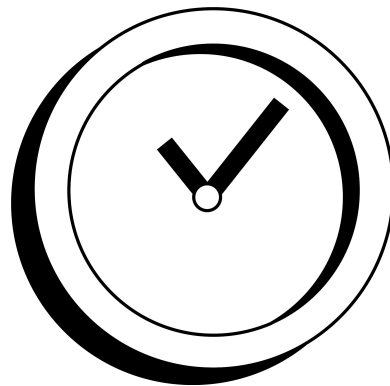


## Задание 4

Работа с API для сбора и обработки данных, а затем создание DataFrame для подготовки данных к анализу.

- Перенести код из блока 3 в Google Colab.
- Модифицировать код, чтобы извлечь адрес, название ресторана, а также координаты (долгота и широта)
- Создать pandas DataFrame из полученных данных

<<20:00-





## Домашнее задание

### 1. Ознакомиться с некоторыми интересными API.

<https://docs.ozon.ru/api/seller/>

<https://developers.google.com/youtube/v3/getting-started>

<https://spoonacular.com/food-api>

### 2. Потренируйтесь делать запросы к API.

Выберите публичный API, который вас интересует, и потренируйтесь делать API-запросы с помощью Postman. Поэкспериментируйте с различными типами запросов и попробуйте получить различные типы данных.

### 3. Сценарий Foursquare

- Напишите сценарий на языке Python, который предложит пользователю ввести интересующую его категорию (например, кофейни, музеи, парки и т.д.).
- Используйте API Foursquare для поиска заведений в указанной категории.
- Получите название заведения, его адрес и рейтинг для каждого из них.
- Скрипт должен вывести название и адрес и рейтинг каждого заведения в консоль.



Вопросы?

Вопросы?



Вопросы?







Спасибо за внимание!