ADA COURSES

HTTP

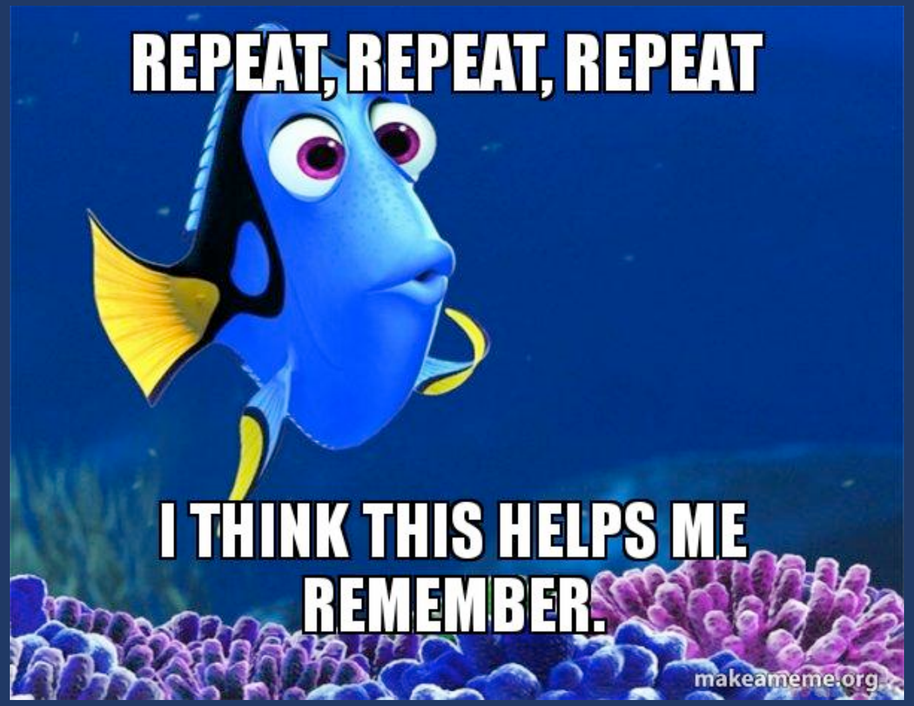
часть 2

CONTENT

- ПОВТОРЕНИЕ
- HTTP ЗАПРОСЫ В PYTHON
- HTTP МЕТОДЫ
- КОДЫ СОСТОЯНИЯ ОТВЕТА HTTP
- ЗАГОЛОВКИ HTTP

ПОВТОРЕНИЕ HTTP

- HTTP (HYPERTEXT TRANSFER PROTOCOL) - ЭТО ПРОТОКОЛ ПЕРЕДАЧИ ДАННЫХ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ПЕРЕДАЧИ ВЕБ-СТРАНИЦ, ИЗОБРАЖЕНИЙ, ВИДЕО И ДРУГИХ ДАННЫХ В ИНТЕРНЕТЕ.
- ОН ЯВЛЯЕТСЯ ОСНОВНЫМ ПРОТОКОЛОМ ДЛЯ ОБМЕНА ДАННЫМИ В ВЕБ-РАЗРАБОТКЕ.
- HTTP РАБОТАЕТ ПО КЛИЕНТ-СЕРВЕРНОЙ МОДЕЛИ, ГДЕ КЛИЕНТ ОТПРАВЛЯЕТ ЗАПРОС НА СЕРВЕР, А СЕРВЕР ОТПРАВЛЯЕТ ОТВЕТ ОБРАТНО КЛИЕНТУ.
- КАЖДЫЙ ЗАПРОС И ОТВЕТ СОДЕРЖИТ ЗАГОЛОВОК И ТЕЛО СООБЩЕНИЯ.
- ЗАГОЛОВОК СОДЕРЖИТ МЕТАДАННЫЕ ЗАПРОСА ИЛИ ОТВЕТА, ТАКИЕ КАК ТИП СОДЕРЖИМОГО, ДАТА, ВРЕМЯ И Т. Д.
- ТЕЛО СООБЩЕНИЯ СОДЕРЖИТ ФАКТИЧЕСКИЕ ДАННЫЕ, ПЕРЕДАВАЕМЫЕ МЕЖДУ КЛИЕНТОМ И СЕРВЕРОМ.



HTTP ЗАПРОСЫ

- **HTTP (HYPERTEXT TRANSFER PROTOCOL)** – ЭТО ПРОТОКОЛ ПЕРЕДАЧИ ГИПЕРТЕКСТА, ИСПОЛЬЗУЕМЫЙ ДЛЯ ПЕРЕДАЧИ ДАННЫХ В ИНТЕРНЕТЕ.
- КЛИЕНТ ОТПРАВЛЯЕТ **HTTP** ЗАПРОС НА СЕРВЕР, А СЕРВЕР ОТПРАВЛЯЕТ **HTTP** ОТВЕТ.
- **HTTP** ЗАПРОС СОСТОИТ ИЗ ТРЕХ ЧАСТЕЙ: МЕТОД, **URL** И ВЕРСИЯ ПРОТОКОЛА.
- **HTTP** ОТВЕТ СОСТОИТ ИЗ ТРЕХ ЧАСТЕЙ: ВЕРСИЯ ПРОТОКОЛА, КОД СОСТОЯНИЯ И СООБЩЕНИЕ.
- **HTTP** ИСПОЛЬЗУЕТ РАЗЛИЧНЫЕ МЕТОДЫ ДЛЯ ОБРАБОТКИ ЗАПРОСОВ, ВКЛЮЧАЯ **GET**, **POST**, **PUT** И **DELETE**.
- **HTTP** ИСПОЛЬЗУЕТ **URL (UNIFORM RESOURCE LOCATOR)** ДЛЯ ОПРЕДЕЛЕНИЯ РЕСУРСА, К КОТОРОМУ СЛЕДУЕТ ОБРАТИТЬСЯ.
- **HTTP** ТАКЖЕ ИСПОЛЬЗУЕТ ЗАГОЛОВКИ ДЛЯ ПЕРЕДАЧИ ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ, ТАКОЙ КАК ТИП СОДЕРЖИМОГО, ДЛИНА СООБЩЕНИЯ И Т.Д.

HTTP МЕТОДЫ

- НАИБОЛЕЕ РАСПРОСТРАНЕННЫЕ МЕТОДЫ **HTTP:**
- **GET -** ПОЛУЧЕНИЕ ДАННЫХ С СЕРВЕРА
- **POST -** ОТПРАВКА ДАННЫХ НА СЕРВЕР ДЛЯ ОБРАБОТКИ
- **PUT -** ОБНОВЛЕНИЕ СУЩЕСТВУЮЩИХ ДАННЫХ НА СЕРВЕРЕ
- **DELETE -** УДАЛЕНИЕ ДАННЫХ НА СЕРВЕРЕ
- ТАКЖЕ СУЩЕСТВУЮТ ДРУГИЕ МЕТОДЫ, ТАКИЕ КАК **HEAD, OPTIONS, PATCH** И **CONNECT,** НО ОНИ МЕНЕЕ РАСПРОСТРАНЕНЫ.

HTTP МЕТОДЫ

GET- ПОЛУЧЕНИЕ ДАННЫХ С СЕРВЕРА

```python
import requests

response = requests.get('https://www.example.com')

print(response.status_code)
print(response.content)

```

HTTP МЕТОДЫ

POST - ОТПРАВКА ДАННЫХ НА СЕРВЕР ДЛЯ ОБРАБОТКИ

```python
import requests

data = {'key1': 'value1', 'key2': 'value2'}
response = requests.post('https://www.example.com', data=data)

print(response.status_code)
print(response.content)

```

HTTP МЕТОДЫ

PUT - ОБНОВЛЕНИЕ СУЩЕСТВУЮЩИХ ДАННЫХ НА СЕРВЕРЕ

```python
import requests

data = {'key1': 'new_value1', 'key2': 'new_value2'}
response = requests.put('https://www.example.com', data=data)

print(response.status_code)
print(response.content)

```

HTTP МЕТОДЫ

DELETE - УДАЛЕНИЕ ДАННЫХ НА СЕРВЕРЕ

```python
import requests

response = requests.delete('https://www.example.com')

print(response.status_code)
print(response.content)

```

HTTP МЕТОДЫ

ОТПРАВКА ЗАГОЛОВКОВ В ЗАПРОСЕ

```python
import requests

headers = {'User-Agent': 'Mozilla/5.0'}
response = requests.get('https://www.example.com', headers=headers)

print(response.status_code)
print(response.content)

```

КОДЫ СОСТОЯНИЯ ОТВЕТА HTTP

- КОДЫ СОСТОЯНИЯ ОТВЕТА HTTP - ЭТО ТРЕХЗНАЧНЫЕ ЧИСЛА, КОТОРЫЕ ПЕРЕДАЮТСЯ В ОТВЕТ НА HTTP ЗАПРОС. КАЖДЫЙ КОД СОСТОЯНИЯ СООБЩАЕТ О СТАТУСЕ ЗАПРОСА И ПОЗВОЛЯЕТ ОПРЕДЕЛИТЬ, БЫЛ ЛИ ЗАПРОС УСПЕШНО ОБРАБОТАН ИЛИ ВОЗНИКЛА ОШИБКА.
- НЕКОТОРЫЕ ПРИМЕРЫ КОДОВ СОСТОЯНИЯ ОТВЕТА HTTP:
  - 200 - OK
  - 201 - CREATED
  - 400 - BAD REQUEST
  - 401 - UNAUTHORIZED
  - 404 - NOT FOUND
  - 500 - INTERNAL SERVER ERROR

```python
import requests

response = requests.get('https://www.example.com')
print(response.status_code)

```

ЗАГОЛОВКИ HTTP

- ЗАГОЛОВКИ HTTP - ЭТО ДОПОЛНИТЕЛЬНЫЕ МЕТАДАННЫЕ, КОТОРЫЕ ПЕРЕДАЮТСЯ ВМЕСТЕ С HTTP ЗАПРОСОМ ИЛИ ОТВЕТОМ.
- НЕКОТОРЫЕ РАСПРОСТРАНЕННЫЕ ЗАГОЛОВКИ HTTP:
- USER-AGENT: ИНФОРМАЦИЯ О БРАУЗЕРЕ ИЛИ КЛИЕНТСКОМ ПРИЛОЖЕНИИ, ОТПРАВЛЯЮЩЕМ ЗАПРОС
- CONTENT-TYPE: MIME-ТИП СОДЕРЖИМОГО, ПЕРЕДАВАЕМОГО В ТЕЛЕ ЗАПРОСА ИЛИ ОТВЕТА
- ACCEPT: MIME-ТИПЫ, КОТОРЫЕ КЛИЕНТ ГОТОВ ПРИНЯТЬ В ОТВЕТ НА ЗАПРОС
- AUTHORIZATION: ИНФОРМАЦИЯ ДЛЯ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ

```python
import requests

url = 'https://www.example.com'
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'}
response = requests.get(url, headers=headers)
print(response.status_code)
print(response.content)

```

ЗАГОЛОВКИ HTTP

- ЗАГОЛОВКИ HTTP - ЭТО ДОПОЛНИТЕЛЬНЫЕ МЕТАДАННЫЕ, КОТОРЫЕ ПЕРЕДАЮТСЯ ВМЕСТЕ С HTTP ЗАПРОСОМ ИЛИ ОТВЕТОМ.
- НЕКОТОРЫЕ РАСПРОСТРАНЕННЫЕ ЗАГОЛОВКИ HTTP:
- USER-AGENT: ИНФОРМАЦИЯ О БРАУЗЕРЕ ИЛИ КЛИЕНТСКОМ ПРИЛОЖЕНИИ, ОТПРАВЛЯЮЩЕМ ЗАПРОС
- CONTENT-TYPE: MIME-ТИП СОДЕРЖИМОГО, ПЕРЕДАВАЕМОГО В ТЕЛЕ ЗАПРОСА ИЛИ ОТВЕТА
- ACCEPT: MIME-ТИПЫ, КОТОРЫЕ КЛИЕНТ ГОТОВ ПРИНЯТЬ В ОТВЕТ НА ЗАПРОС
- AUTHORIZATION: ИНФОРМАЦИЯ ДЛЯ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ

```python
import requests

url = 'https://www.example.com'
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'}
response = requests.get(url, headers=headers)
print(response.status_code)
print(response.content)

```

HOUSTON, WE HAVE A PROBLEM

1. Отправьте GET запрос на страницу поиска Google и выведитет в консоль код ответа.
2. Создайте простой веб-сервер на Python с помощью модуля http.server и отобразите на нем страницу с текстом "Hello, World!".
3. Используя библиотеку requests, отправьте POST запрос на сайт httpbin.org/post с данными в формате JSON и выведите ответ.
4. Напишите скрипт на Python, который будет отправлять GET запрос на сайт httpbin.org/robots.txt и выводить содержимое файла robots.txt.
5. Напишите программу на Python, которая будет загружать изображение с сайта и сохранять его на жестком диске.
6. Напишите скрипт на Python, который будет проверять доступность сайта каждые 10 минут и отправлять уведомление
7. Напишите программу на Python, которая будет отправлять POST запрос на сайт httpbin.org/redirect/1 и выведите ответ, который будет содержать информацию о редиректе.
8. Напишите скрипт на Python, который будет отправлять GET запрос на сайт httpbin.org/get с параметрами в URL строке и выведите ответ.

1. Напишите программу на Python, которая будет отправлять PUT запрос на сайт httpbin.org/put с данными в формате JSON и выведите ответ.
2. Напишите скрипт на Python, который будет отправлять GET запрос на сайт httpbin.org/image/png и отображать полученное изображение в окне.
3. Используя библиотеку requests, отправьте PUT запрос на сайт httpbin.org/put с данными в формате HTML и выведите ответ.
4. Напишите программу на Python, которая будет отправлять DELETE запрос на сайт httpbin.org/delete с параметрами в URL строке и выведите ответ.
5. Напишите скрипт на Python, который будет отправлять POST запрос на сайт httpbin.org/post с файлом вложения и выведите ответ.
6. Напишите программу на Python, которая будет отправлять GET запрос на сайт httpbin.org/xml и выводить содержимое ответа в формате XML.

СДЕЛАЙ ЕСЛИ СМОЖЕШЬ

1. Отправьте POST запрос на страницу аутентификации с использованием библиотеки requests.
2. Отправьте GET запрос на API сайта OpenWeatherMap, получает текущую погоду для заданного города и выводит ее в консоль в виде json.
3. Отправьте POST запрос на API сайта Trello для создания новой задачи.
4. Написать программу, который возвращает список постов при GET запросе и создает новый пост при POST запросе.
5. Написать программу на Python, которая использует библиотеку requests для отправки GET запроса на API сайта GitHub, получает список репозиториев пользователя и выводит его в консоль.
6. Отправьте POST запрос на API сайта GitHub для создания нового репозитория.
7. Отправьте GET запрос на API сайта Reddit, получите список популярных постов на определенной теме и выводит его в консоль.
8. **Написать программу на Python, которая использует библиотеку requests для отправки POST запроса на API сайта Twitter для отправки твита.**
9. Используйте **библиотеку requests для отправки GET запроса на API сайта Wikipedia, получает статью по определенной теме и выводит ее в консоль.**
10. **Отправьте POST запроса на API сайта Facebook для создания нового поста.**