Ответы на вопросы 21 лаба

1. **Дайте определение понятию «Протокол HTTP».**
   1. **HTTP** – это **протокол** передачи информации в интернете, который расшифровывается как «**протокол** передачи гипертекста» (HyperText Transfer **Protocol**).
2. **Какая библиотека используется для получения запросов HTTP.**
   1. Библиотека requests
3. **Какая технология используется для протокола HTTP.**
   1. Основой **HTTP** является **технология** «клиент-сервер» - предполагается, что существует множество потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом
4. **Какой метод используется для попытки извлечь данные из определенного ресурса?**
   1. .get
5. **Дайте определение понятию «HTTP коды состояний».**
   1. Самыми первыми данными, которые будут получены через Response, будут коды состояния. **Коды состояния** сообщают о статусе запроса
6. **Перечислите HTTP коды состояний.**
   1. [102 Processing](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#102) («идёт обработка»).
   2. [200 OK](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#200) («хорошо»);
   3. [204 No Content](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#204) («нет содержимого»);
   4. [400 Bad Request](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#400) (неверный запрос»);
   5. [404 Not Found](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#404) («не найдено»);
   6. [408 Request Timeout](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#408) («истекло время ожидания»);
   7. [410 Gone](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#410) («удалён»);
   8. [500 Internal Server Error](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#500) («внутренняя ошибка сервера»);
   9. [521 Web Server Is Down](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#521) («веб-сервер не работает»);
   10. [522 Connection Timed Out](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#522) («соединение не отвечает»);
   11. [524 A Timeout Occurred](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP#524) («время ожидания истекло»);
7. **Для чего используется функция .status\_code?**
   1. Используя .status\_code, можно увидеть код состояния, который возвращается с сервера. print(response.status\_code)

Ответ: 200

status\_code вернул значение 200. Это значит, что запрос был выполнен успешно, а сервер ответил, отобразив запрашиваемую информацию.

1. **Как получить содержимое страницы в Requests?**
   1. Для того, чтобы получить содержимое запроса в байтах, необходимо использовать **.content**
2. **Как получить HTTP заголовки в Requests?**
   1. Для просмотра HTTP заголовков используется атрибут .headers.

import requests

response = requests.get('https://api.github.com')

print(response.headers) **.headers** возвращает словарь, что позволяет получить доступ к значению заголовка HTTP по ключу. Например, для просмотра типа содержимого ответного пейлоада, требуется использовать Content-Type.