**Задание 04.** Ответьте на следующие вопросы.

**response**.**writeHead**(StatusCode[, statusMessage][, headers]). Параметры: Этот метод принимает следующий аргумент в качестве параметра.

1. Дайте определение понятию «Интернет».

Всемирная информационная компьютерная сеть, связывающая между собой как пользователей компьютерных сетей, так и пользователей индивидуальных компьютеров для обмена информацией.

1. Дайте определение понятию «Служба Интернет».

Веб-**служба**, веб-сервис (англ. **web service**) — идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.

1. Дайте определение понятию «Узел сети Интернет».

устройство, имеющее IP-адрес и подключенное сети Интернет (обычно к сети Интернет-провайдера).

1. Дайте определение понятию «клиент-серверное приложение».

приложение (программа) с клиент-серверной архитектурой: приложение, состоящее из двух компонент – клиента и **сервера**; клиент и сервер взаимодействуют между собой в соответствии с заданными правилами (спецификациями, **протоколами**); для взаимодействия между клиентом и сервером в соответствии с правилами (спецификацией, протоколом) должно быть установлено **соединение**; **инициатором соединения всегда является клиент**.

1. Дайте определение понятию «сетевой протокол».

это набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть компьютерами.

1. Перечислите основные свойства протокола HTTP.

**HTTP:** основные свойства

* версии HTTP/1.1 – действующий (текстовый), HTTP/2 – черновой (не распространен, бинарный);
* два типа абонентов: клиент и сервер;
* два типа сообщений: request и response;
* от клиента к серверу – request;
* от сервера к клиенту – response;
* на один request всегда один response, иначе ошибка;
* одному response всегда один request, иначе ошибка;
* TCP-порты: 80, 443;
* для адресации используется URI или URN;
* поддерживается W3C, описан в нескольких RFC.

1. Перечислите состав информации, пересылаемой в HTTP-запросе.

**Request**:

* метод;
* URI;
* версия протокола (HTTP/1.1);
* заголовки (пары: имя/заголовок);
* параметры (пары: имя/заголовок);
* расширение.

1. Перечислите состав информации, пересылаемой в HTTP-ответе.

**Response:**

* версия протокола (HTTP/1.1);
* код состояния (1xx, 2xx, 3xx, 4xx, 5xx);
* **1xx**: информационные сообщения;
* **2xx**: успешный ответ;
* **3xx**: переадресация;
* **4xx**: ошибка клиента;
* **5xx**: ошибка сервера.
* пояснение к коду состояния;
* заголовки (пары: имя/заголовок);
* расширение.

1. Дайте определение понятию «web-приложение».

[клиент-серверное](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_%E2%80%94_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) приложение, в котором [клиент](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) взаимодействует с [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) при помощи [браузера](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80).

1. Дайте определение понятиям «frontend» и «backend».

**frontend**) — клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса. Бэкэнд (англ. ... Фронт- и бэкэнд — **это** вариант архитектуры программного обеспечения(сервер)

1. Дайте определение понятию «кроссплатформенное приложение».

приложение, способное работать на более чем одной программно-аппаратной (аппаратура + операционная система) платформе. Кроссплатформенность может быть достигнута различными способами: 1) на уровне компилятора (С, С++); 2) на уровне среды (фреймворка) исполнения (Java/JVM, C#/.NET CORE, JS/Node.js/V8).

1. Изобразите и поясните общую схему web-приложения.











1. Назовите основные технологии разработки серверных кроссплатформенных приложений.

технологии для разработки кроссплатформенных web-серверов:

- PHP/Apache(апач), LAMP;

- Java/JVM/Application Server;

- C#/ASP.NET CORE;

- Python/Django;

- Ruby on Rails;

- JS/Node.js, ….

1. Поясните понятие «асинхронная операция».

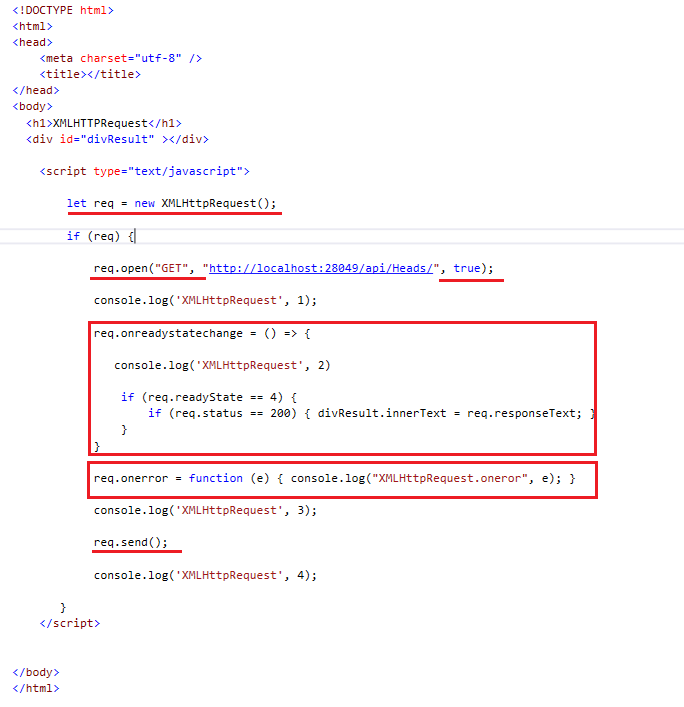
операция, при которой поток, выдавший http-запрос, не блокируется до поступления ответа; для обработки ответа применяется функция обратного вызова.

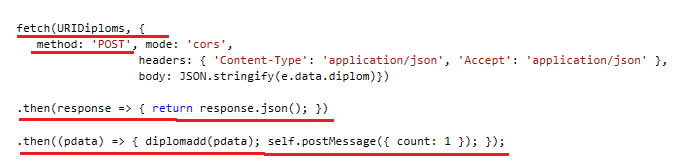
операция называется асинхронной, если ее выполнение осуществляется в 2 фазы: 1) заявка на исполнение; 2) получение результата; при этом участвуют два механизма: A-механизм, формирующий заявку и потом получающий результат; B-механизм, получающий заявку от A, исполняющий операцию и отправляющий результат A; продолжительность исполнения операции B-механизмом, как правило, непредсказуемо; в то время пока B-механизм исполняет операцию, А-механизм выполняет собственную работу. Применение асинхронности не противоречит применению многопоточности.

1. Поясните принцип выполнения асинхронного запроса с помощью объекта XMLHTTPRequest и Fetch.

AJAX: Asynchronous JavaScript and XML – асинхронный JavaScript and XML – методология (подход) построения динамических приложений, при которых не осуществляется полная перезагрузка html-страниц.







1. Поясните основное назначение сервера NODE.JS.

программная платформа для разработки серверных web-приложений на языке JS/V8.

1. Перечислите основные свойства сервера NODE.JS.

основные свойства:

* основан на **Chrome V8**;
* **среда (контейнер) исполнения** приложений на JavaScript;
* поддерживает механизм **асинхронности**;
* ориентирован на **события**;
* **однопоточный** (код приложения исполняется только в одном потоке, один стек вызовов); обычно в серверах для каждого соединения создается свой поток, в Node.js все соединения обрабатываются в одном JS-потоке;
* **не блокирует** выполнение кода при вводе/выводе (в файловой системе до 4х одновременно);
* в состав Node.js входят инструменты: **npm** – пакетный менеджер; **gyp** - Python-генератор проектов; **gtest** – Google фреймворк для тестирования С++ приложений;
* использует библиотеки: **V8** – библиотека V8 Engine, **libuv** – библиотека для абстрагирования неблокирующих операций ввода/вывода; **http-parser** – легковесный парсер http-сообщений (написан на C и не выполняет никаких системных вызовов); **c-ares** -библиотека для работы с DNS; **OpenSSL** – библиотека для криптографии; **zlib** – сжатие и распаковка.