1. **Дайте определение понятию «Интернет».**

**Интернет** — всемирная сеть, построенная на стеке протоколов TCP/IP.

1. **Дайте определение понятию «Служба Интернет».**

Службы интернет — **сервер + протокол** / протокол, который описывает, как работать с этим сервером. Стандартные серверы, которые *прослушивают хорошо известные порты (от 0 до 1024*).В определении говорим сервер + ПРОТОКОЛ, а как характерное свойство говорим уже про порты

1. **Дайте определение понятию «Узел сети Интернет».**

**узел Интернет:** устройство, имеющее IP-адрес и подключенное сети Интернет (обычно к сети Интернет-провайдера). Каждый узел характеризуется своей программно-аппаратной платформой – аппаратурой и операционной системой.

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

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».**

Сетевой протокол — это **набор правил и соглашений**, которые определяют порядок и формат обмена данными между компьютерами, устройствами или программами в сети.

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);
* пояснение к коду состояния;
* заголовки (пары: имя/заголовок);
* расширение.

1. **Дайте определение понятию «web-приложение».**

клиент-серверное приложение, у которого клиент и сервер взаимодействуют по протоколу HTTP называется **web-приложением**.

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

когда говорят о разработке web-приложения, говорят о разработке **frontend** (клиента) и **backend** (сервера)

1. **Дайте определение понятию «кроссплатформенное приложение».**

**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.**
2. **Архитектура web-приложения:**

****

****

****

****

****

1. **Назовите основные технологии разработки серверных кроссплатформенных приложений.**

Для разработки **серверных кроссплатформенных приложений** используют технологии и платформы, которые работают на разных операционных системах (Windows, Linux, macOS).

**🔹 Основные технологии:**

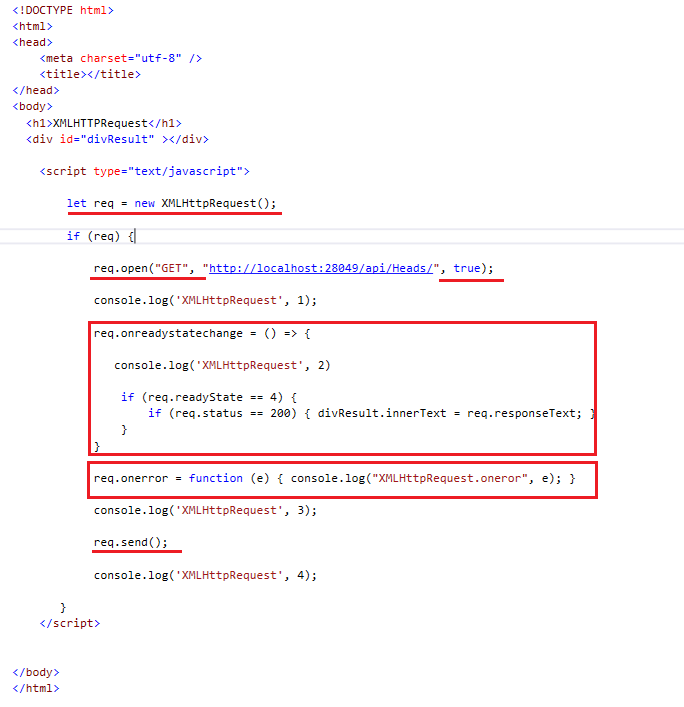
1. **Node.js** – серверная платформа на JavaScript, часто используется с Express.js, NestJS.
2. **Java + Spring Framework** – кроссплатформенные приложения на языке Java.
3. **.NET Core / ASP.NET Core** – современная кроссплатформенная версия .NET для C#.
4. **Python** – с фреймворками Django, Flask, FastAPI.
5. **PHP** – с веб-фреймворками Laravel, Symfony.
6. **Ruby on Rails** – кроссплатформенный фреймворк на Ruby.
7. **Go (Golang)** – язык Google для высоконагруженных серверных приложений.
8. **C++ с кроссплатформенными библиотеками** (например, Boost, Poco).
9. **Поясните понятие «асинхронная операция».**

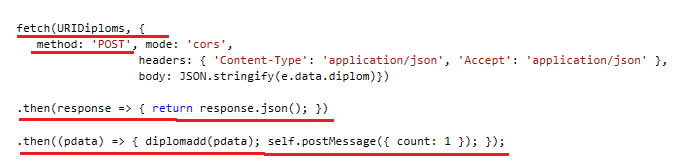
**Понятие асинхронности**: операция называется асинхронной, если ее выполнение осуществляется в 2 фазы: 1) заявка на исполнение; 2) получение результата; при этом участвуют два механизма: A-механизм, формирующий заявку и потом получающий результат; B-механизм, получающий заявку от A, исполняющий операцию 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. **Асинхронные операции ввода вывода в операционной системе:**



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

****

****

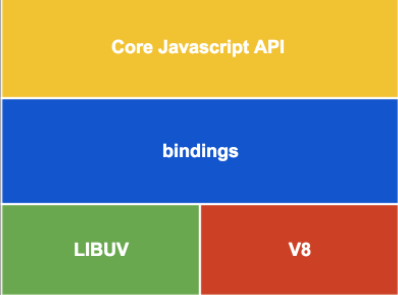
**??**

1. **Поясните основное назначение сервера NODE.JS.**

* основная сфера применения: **разработка web-серверов**;
* **NODEJS:** программная платформа для разработки серверных web-приложений на языке JS/V8.

1. **Перечислите основные свойства сервера NODE.JS.**
2. **NODEJS:** основные свойства:

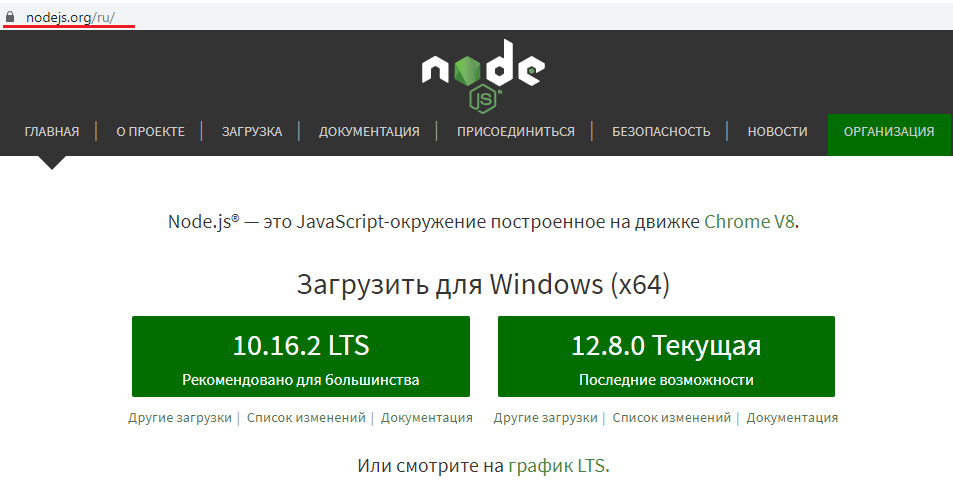
* основан на **Chrome V8**;
* **среда (контейнер) исполнения** приложений на JavaScript;
* поддерживает механизм **асинхронности**;
* ориентирован на **события**;
* **однопоточный** (код приложения исполняется только в одном потоке, один стек вызовов); обычно в серверах для каждого соединения создается 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.



* разработчик: **Райан Дал**:

****

* первая версия: **2009 г**.;
* стабильные версии: с **2015 г., Node.js 4.0.0;**
* официальный сайт:

****

* основная сфера применения: **разработка web-серверов**;
* четные версии Node.js –версии длительной поддержки (LST), нечетные версии – нестабильные версии, включающие последние разработки.