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

**Internet** — это глобальная компьютерная сеть, которая по протоколу TCP/IP объединяет сети различных регионов, государств, научных организаций в одну общую сеть.

# 2. Дайте определение понятию «Служба Интернет».

**Служба Интернет** - подразумевает наличие сервера и протокола для доступа к нему.

# 3. Дайте определение понятию «Узел сети Интернет».

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

# 4.Дайте определение понятию «клиент-серверное приложение».

**Клиент-серверное приложение** - Приложение с клиент-серверной архитектурой: состоящее из двух компонентов: **клиент и сервер**; Клиент и сервер взаимодействуют между собой в соответствии с заданными правилами(спецификациями и протоколами); Для взаимодействия клиента с сервером должно быть установлено соединение; Инициатором соединения всегда является клиент.

# 5. Дайте определение понятию «сетевой протокол».

**Сетевой протокол** - это набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включенными в сеть компьютерами.

# 6. Перечислите основные свойства протокола HTTP.

- 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.

# 7. Перечислите состав информации, пересылаемой в HTTP-запросе.

**Request:**

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

# 8. 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.

# 9. Дайте определение понятию «web-приложение».

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

# 10. Дайте определение понятиям «frontend» и «backend».

**frontend** - клиентская сторона клиент-серверного приложения

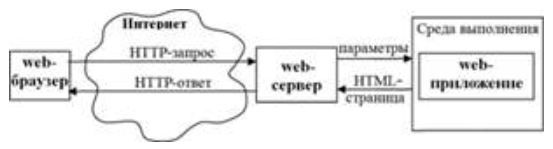
**backend** - серверная сторона

# 11. 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)

# 12. Изобразите и поясните общую схему web-приложения.



Пользователь с помощью браузера отправляет HTTP-запрос по определенному URL-адресу, который сопоставляется с ресурсами на web-сервере. Если данный URL-адрес указывает на динамический ресурс (web-приложений), то сервер запускает некоторую внешнюю программу (web-приложение), формирующую HTML-страницу, которая может быть показана браузером, и возвращает ее клиенту. Основной частью web-приложения является его логика на стороне web-сервера.

# 13. Назовите основные технологии разработки серверных кроссплатформенных приложений.

* PHP/Apache, LAMP
* Java/JVM/Application Server
* C#/ASP.Net Core
* Python/Django
* Ruby on Rails
* JS/Node.js

# 14. Поясните понятие «асинхронная операция».

Операция называется асинхронной, если ее выполнение осуществляется в 2 фазы:

1. заявка на исполнение
2. получение результата

# -----15. Поясните принцип выполнения асинхронного запроса с помощью объекта XMLHTTPRequest и Fetch.

**XMLHttpRequest** – это встроенный в браузер объект, который даёт возможность делать HTTP-запросы к серверу без перезагрузки страницы.

1. Создать **XMLHttpRequest**.
2. Инициализировать его. **method** – HTTP-метод. Обычно это "GET" или "POST". **URL** – URL, куда отправляется запрос: строка, может быть и объект [URL](https://learn.javascript.ru/url). **async** – если указать false, тогда запрос будет выполнен синхронно
3. Послать запрос.
4. Слушать события, чтобы получить ответ.

**Fetch** возвращает **Promise** (обещание - которое будет выполнено или отклонено). Для того чтобы получить данные из промиса, нам нужно включить в цепочку два колбека с помощью метода then.

* Делаем запрос, получаем промис
* После того как промис выполнен успешно, получаем специальный объект ответа.

Выполним метод объекта ответа json(), чтобы получить данные.

* Метод json() возвращает промис, так что далее снова вызываем then и в колбеке выводим данные в консоль

fetch('https://jsonplaceholder.typicode.com/posts/1')

.then(response => response.json())

.then(data => console.log(data));

# 16. Поясните основное назначение сервера NODE.JS.

Для разработки серверных web-приложений на языке JS/V8

# 17. Перечислите основные свойства сервера NODE.JS.

* основан на Chrome V8;
* среда исполнения приложений на JavaScript;
* поддерживает механизм асинхронности;
* ориентирован на события;
* однопоточный
* не блокирует выполнение кода при вводе/выводе;