1. Что такое **асинхронный запрос**?

**Асинхронный запрос** - это запрос к серверу или выполнение операции, которое не блокирует исполнение кода и позволяет продолжить выполнение других задач в то время, как запрос обрабатывается. В контексте веб-приложений асинхронные запросы часто используются для обмена данными между браузером и сервером без блокировки пользовательского интерфейса. Это позволяет создавать отзывчивые веб-приложения.

1. Что такое **AJAX**?

**AJAX** (Asynchronous JavaScript and XML) - это набор технологий и методов, используемых для создания асинхронных веб-приложений. Вместо того, чтобы перезагружать всю веб-страницу, AJAX позволяет обмениваться данными с сервером в фоновом режиме и обновлять только части страницы. AJAX использует JavaScript для отправки асинхронных запросов на сервер и обработки ответов.

1. Поясните назначение HTTP-заголовка **Content-Type**

**HTTP-заголовок Content-Type** указывает тип контента, который передается в теле HTTP-запроса или ответа. Этот заголовок сообщает браузеру или серверу, как правильно интерпретировать данные. Например, когда браузер получает ответ с заголовком Content-Type: text/html, он знает, что это HTML-контент и должен отобразить его как веб-страницу. Content-Type также может указывать на JSON, XML, изображения и другие форматы данных

1. Поясните назначение функции **require**

**Функция require** в Node.js используется для импорта модулей и библиотек в вашем коде. Она позволяет вам подключать внешние модули, встроенные модули Node.js или пользовательские модули, которые вы создали. Функция require возвращает экспортированные объекты или функции из модуля, что позволяет вам использовать их в вашем коде

1. Поясните понятие «**Модуль Node.js**»

**Модуль Node.js** - это независимый и переиспользуемый фрагмент кода, который может быть подключен к вашему приложению с помощью функции require. Модули позволяют организовать и структурировать ваш код, улучшить его читаемость и облегчить сопровождение приложения

1. Поясните понятие «**Node.js built-in modules**» («**Node.js Core modules**»)

**Node.js built-in modules** (или **Node.js Core modules**) - это модули, которые поставляются с Node.js и доступны для использования без дополнительной установки. Эти модули предоставляют базовую функциональность для работы с файлами, сетью, HTTP, событиями и другими аспектами разработки приложений на Node.js. Примеры встроенных модулей включают fs (для работы с файлами), http (для работы с HTTP-серверами и клиентами), и events (для работы с событиями)

1. Какой модуль **NODE.JS** обеспечивают работу с протоколом HTTP, с файловой системой

Для работы с протоколом HTTP и файловой системой в Node.js используются следующие встроенные модули:

* **http** - для создания и работы с HTTP-серверами и клиентами.
* **https** - аналогичен модулю http, но для безопасных HTTPS-соединений.
* **fs** (File System) - для работы с файлами и директориями на файловой системе

1. Что такое **npm?**

**npm** (Node Package Manager) - это менеджер пакетов для Node.js. Он позволяет разработчикам управлять зависимостями и устанавливать пакеты (библиотеки и модули), которые используются в Node.js-проектах. npm также предоставляет возможность публикации и обмена своими пакетами с сообществом разработчиков Node.js