1. Асинхронный запрос - это запрос к серверу или выполнение операции, который не блокирует выполнение остального кода программы. Вместо того, чтобы ждать завершения запроса или операции, программа может продолжать выполнять другие задачи. Когда асинхронная операция завершится, будет вызвана соответствующая функция обратного вызова или обработано событие, чтобы обработать результат запроса или выполненной операции.
2. AJAX (Asynchronous JavaScript and XML) - это набор технологий, который позволяет выполнять асинхронные запросы к серверу из браузера. Вместо перезагрузки всей веб-страницы для получения данных или обновления содержимого, AJAX позволяет загружать данные и взаимодействовать с сервером без изменения текущей страницы. Он использует JavaScript для отправки запросов на сервер и обработки полученных данных, и может работать с разными форматами данных, не обязательно XML.
3. HTTP-заголовок Content-Type используется для указания типа данных, передаваемых в HTTP-запросе или ответе. Этот заголовок сообщает браузеру или серверу, как обрабатывать тело запроса или ответа. Примеры типов данных, указываемых в заголовке Content-Type, включают "text/html" для HTML-документов, "application/json" для JSON-данных и "image/jpeg" для изображений. Это позволяет корректно интерпретировать содержимое HTTP-сообщения.
4. Функция **require** - это встроенная функция в Node.js, используемая для импорта модулей или файлов в Node.js приложении. Она позволяет подключать внешние модули или файлы, чтобы использовать их функциональность в вашем коде. Пример использования **require**:

const fs = require('fs'); // Импорт модуля для работы с файловой системой

1. Модуль Node.js - это самостоятельный кусок кода, который содержит функции, переменные и другие объекты, предназначенные для определенной цели или функциональности. Модули помогают организовать и структурировать код в Node.js приложении, делая его более модульным и понятным. Модули могут быть созданы как встроенные (входящие в состав Node.js) так и сторонние, устанавливаемые с использованием npm.
2. Node.js built-in modules (Node.js Core modules) - это набор модулей, которые поставляются вместе с Node.js и доступны по умолчанию без необходимости установки. Они включают в себя модули для работы с файловой системой, сетевыми запросами, работой с потоками и другие базовые функциональности. Примеры встроенных модулей в Node.js включают **fs** (для работы с файлами), **http** (для работы с HTTP-серверами), и **os** (для работы с операционной системой).
3. Для работы с протоколом HTTP в Node.js используется встроенный модуль **http**. Этот модуль позволяет создавать HTTP-серверы и отправлять HTTP-запросы. Для работы с файловой системой в Node.js также используется встроенный модуль **fs**, который предоставляет функции для чтения и записи файлов.
4. npm (Node Package Manager) - это менеджер пакетов для Node.js, который позволяет разработчикам управлять зависимостями и устанавливать сторонние модули и библиотеки для своих Node.js проектов. npm также предоставляет множество команд для управления пакетами, включая установку, обновление, удаление и публикацию пакетов. npm - это ключевой инструмент в экосистеме Node.js для облегчения разработки и совместного использования кода.