1. [Интернет - это глобальная сеть компьютерных систем, которая позволяет пользователям обмениваться информацией и общаться друг с другом](https://fb.ru/article/41971/a-vyi-znaete-kak-ustroen-internet" \t "_blank).
2. Служба Интернет - это услуга, которая предоставляет пользователям доступ к Интернету и его ресурсам. Это может включать в себя предоставление подключения к Интернету, веб-хостинг, электронную почту и другие услуги.
3. [Узел сети Интернет - это устройство, соединенное с другими устройствами как часть компьютерной сети Интернет2](https://ru.wikipedia.org/wiki/%D0%A3%D0%B7%D0%B5%D0%BB_%D1%81%D0%B5%D1%82%D0%B8). [Узлами могут быть компьютеры, мобильные телефоны, карманные компьютеры, а также специальные сетевые устройства, такие как маршрутизаторы, коммутаторы или концентраторы](https://ru.wikipedia.org/wiki/%D0%A3%D0%B7%D0%B5%D0%BB_%D1%81%D0%B5%D1%82%D0%B8).
4. [Клиент-серверное приложение - это тип приложения, в котором клиентское программное обеспечение запрашивает услуги или ресурсы у серверного программного обеспечения](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). Клиент и сервер могут находиться на разных компьютерах и взаимодействовать через компьютерную сеть.
5. [Сетевой протокол - это набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включенными в сеть устройствами](https://dic.academic.ru/dic.nsf/ruwiki/1089).
6. Основные свойства протокола HTTP:
   * HTTP является протоколом клиент-серверного взаимодействия.
   * HTTP используется для передачи гипертекстовых документов, таких как HTML.
   * HTTP является протоколом без сохранения состояния.
   * HTTP может использовать любой другой протокол транспортного уровня с гарантированной доставкой.
   * HTTP может быть использован для получения только частей документа с целью обновления веб-страницы по запросу.
7. Состав информации, пересылаемой в HTTP-запросе:
   * Метод запроса (GET, POST и т.д.)
   * URL-адрес запрашиваемого ресурса
   * Версия протокола HTTP
   * Заголовки запроса (например, User-Agent, Host и Accept)
   * Тело запроса (для методов POST и PUT)
8. Состав информации, пересылаемой в HTTP-ответе:
   * Версия протокола HTTP
   * Код состояния (200, 404 и т.д.)
   * Фраза состояния (ОК, Not Found и т.д.)
   * Заголовки ответа (например, Content-Type, Content-Length и Date)
   * Тело ответа (содержимое запрашиваемого ресурса)
9. Группы заголовков HTTP:
   * Запросные заголовки (Request headers) содержат дополнительную информацию о запросе клиента.
   * Заголовки ответа (Response headers) содержат дополнительную информацию о ответе сервера.
   * Заголовки представления содержимого (Representation headers) содержат информацию о теле сообщения.
   * Заголовки полезной нагрузки (Payload headers) содержат представление данных полезной нагрузки.
   * Заголовки управления кэшированием (Caching headers) определяют параметры кэширования ответов.
10. [Web-приложение - это программное обеспечение, которое работает на удаленном сервере и доступно через браузер пользователя](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5). Web-приложения могут включать в себя такие функции, как обработка форм, онлайн-магазины, социальные сети и многое другое.
11. Frontend - это часть web-приложения, которая отвечает за взаимодействие с пользователем. Она включает в себя пользовательский интерфейс и обеспечивает отображение информации на экране пользователя. Backend - это часть web-приложения, которая отвечает за обработку данных и взаимодействие с базой данных. Она работает на сервере и обеспечивает функционирование приложения.
12. Кроссплатформенное приложение - это программное обеспечение, которое может работать на нескольких операционных системах или платформах без необходимости изменения кода.
13. Общая схема web-приложения включает в себя клиентскую часть (frontend), которая работает в браузере пользователя и отвечает за отображение информации и взаимодействие с пользователем, и серверную часть (backend), которая работает на удаленном сервере и отвечает за обработку данных и взаимодействие с базой данных. Клиентская и серверная части взаимодействуют друг с другом через протокол HTTP.
14. Основные технологии разработки серверных кроссплатформенных приложений включают в себя языки программирования, такие как Java, Python, Ruby, PHP, JavaScript (Node.js) и другие, а также фреймворки для разработки web-приложений, такие как Ruby on Rails, Django, Express и другие.
15. Асинхронная операция - это операция, которая выполняется параллельно с другими операциями без блокирования выполнения программы. Это позволяет улучшить производительность приложения за счет параллельного выполнения задач.
16. Node.js - это кроссплатформенная среда выполнения JavaScript, которая позволяет использовать JavaScript на стороне сервера для создания масштабируемых web-приложений.
17. Основные свойства Node.js:
    * Кроссплатформенность: Node.js может работать на разных операционных системах.
    * Масштабируемость: Node.js позволяет создавать масштабируемые web-приложения благодаря асинхронной архитектуре.
    * Большое сообщество: Node.js имеет большое и активное сообщество разработчиков, которые создают новые модули и инструменты для улучшения работы с Node.js.
    * Широкий выбор модулей: Node.js имеет большое количество модулей для решения различных задач.
    * Простота использования: Node.js легко устанавливается и имеет простой синтаксис для создания web-приложений.