* 9. Интернет - это мировая система связанных компьютерных сетей, использующая стандартный протокол передачи данных (TCP/IP) для передачи информации и данных пользователей.  
    
  10. Служба Интернет - это услуга или функционал, предоставляемый через Интернет, например, электронная почта, веб-серфинг или облачные сервисы.  
    
  11. Узел сети Интернет - это активное сетевое устройство или сервер, принимающий, передающий или обрабатывающий информацию в Интернет.  
    
  12. Клиент-серверное приложение - это программная архитектура, в которой клиент (обычно пользовательский интерфейс) делает запросы к серверу, а сервер отвечает, предоставляя необходимую информацию или услуги.  
    
  13. Сетевой протокол - это набор правил и стандартов, которые определяют, как данные передаются и получаются через сетевые соединения.  
    
  14. Основные свойства протокола HTTP:  
    
  Безсостояние (stateless): каждый запрос от клиента к серверу рассматривается как новый.  
  Основан на TCP/IP.  
  Поддерживает методы запросов, такие как GET, POST, PUT, DELETE.  
  Прост в использовании и расширяем.  
  15. Информация, пересылаемая в HTTP-запросе:  
    
  Метод запроса (например, GET или POST).  
  URL.  
  Версия HTTP.  
  Заголовки (например, User-Agent, Accept).  
  Тело запроса (не всегда присутствует).  
  16. Информация, пересылаемая в HTTP-ответе:  
    
  Статус-код (например, 200 OK или 404 Not Found).  
  Версия HTTP.  
  Заголовки ответа (например, Server, Content-Type).  
  Тело ответа.  
  17. Группы заголовков HTTP:  
    
  Общие заголовки (General headers) — например, Date, Cache-Control.  
  Заголовки запроса (Request headers) — например, Accept, User-Agent.  
  Заголовки ответа (Response headers) — например, Server, WWW-Authenticate.  
  Заголовки сущности (Entity headers) — например, Content-Type, Content-Length.  
  18. Web-приложение - это приложение, доступное через веб-браузер посредством сети.  
    
  19.  
    
  Frontend - это часть приложения, ответственная за интерфейс и взаимодействие с пользователем.  
  Backend - это серверная часть приложения, обеспечивающая логику, обработку данных и взаимодействие с базами данных.  
  20. Кроссплатформенное приложение - это программное обеспечение, разработанное для работы на нескольких платформах или операционных системах.  
    
  21. Общая схема web-приложения: пользовательский интерфейс (frontend) взаимодействует с сервером (backend) посредством HTTP-запросов. Сервер обрабатывает эти запросы, может обращаться к базе данных и возвращает результаты обратно на frontend.  
    
  22. Основные технологии разработки серверных кроссплатформенных приложений: Node.js, .NET Core, Java (Spring), Python (Flask, Django), Ruby on Rails, Go.  
    
  23. Асинхронная операция - это операция, которая выполняется параллельно с основным потоком выполнения, что позволяет основному потоку продолжать выполнение без ожидания завершения этой операции.  
    
  24. Основное назначение NODE.JS: Node.js - это среда выполнения JavaScript, позволяющая выполнять код на стороне сервера. Она используется для создания масштабируемых серверных приложений.  
    
  25. Основные свойства NODE.JS:  
    
  Основан на V8 JavaScript Engine от Google.  
  Асинхронное и событийно-ориентированное программирование.  
  Быстрое выполнение кода.  
  Большой экосистемой пакетов (npm).  
  Кроссплатформенность.