

Основы WEB-программирования

02.03.03 -Математическое обеспечение и администрирование информационных систем,
направленность (профиль) -разработка и администрирование информационных систем

<https://vikchas.ru>

Вопросы 2-ой контрольной точки 28-30.10.2025

Часовских Виктор Петрович
доктор технических наук, профессор кафедры
ШИиКМ, ФГБОУ ВО «Уральский государственный
экономический университет

Екатеринбург 2025

HTML и структура документа

1. Что такое HTML и какова его роль в веб-разработке?

- HTML (HyperText Markup Language) — это язык разметки для создания структуры веб-страниц
- Определяет структуру и семантическое содержание страницы
- Является основой для построения веб-документов

2. Объясните структуру стандартного HTML5-документа.

HTML

```
<!DOCTYPE html>
```

```
<html lang="ru">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Заголовок страницы</title>
```

```
</head>
```

```
<body>
```

```
  <!-- Содержимое страницы -->
```

```
</body>
```

```
</html>
```

3. Что такое семантический HTML и почему он важен?

- Семантический HTML использует теги, отражающие смысл контента (например,,,)
- Улучшает доступность для людей с ограниченными возможностями
- Помогает поисковым системам лучше индексировать страницы
- Улучшает читаемость кода и его поддержку

4. Как осуществляется валидация HTML-документа и почему это важно?

- Валидация проверяет соответствие кода техническим стандартам W3C

- Можно использовать онлайн-инструменты (W3C Validator) или локальные программы
- Помогает выявить ошибки, которые могут вызвать проблемы с отображением в разных браузерах

CSS и стилизация

5. Что такое CSS и как он связан с HTML?

- CSS (Cascading Style Sheets) — язык для описания внешнего вида HTML-документа
- Позволяет отделить представление от структуры и содержимого
- Подключается к HTML с помощью тегов, или через атрибут style

6. Какие существуют способы подключения CSS к веб-странице?

- Внешняя таблица стилей:
- Встроенные стили: в
- Инлайн-стили: атрибут style="property: value"
- Импорт: @import url('styles.css')(внутри CSS-файла или стилевого тега)

7. Объясните разницу между классами, идентификаторами и тегами для селекторов в CSS.

- Теги (например, div, p) — выбирают все элементы определенного типа
- Классы (.classname) — могут применяться к множеству элементов на странице
- Идентификаторы (#id) — должны быть уникальными на странице
- Специфичность: ID > класс > тег

8. Что такое Box Model в CSS и какие свойства он включает?

- Это модель, описывающая прямоугольные боксы для каждого HTML-элемента
- Состоит из content (содержимое), padding (внутренний отступ), border (граница) и margin (внешний отступ)
- Размер бокса зависит от свойства box-sizing (content-box или border-box)

JavaScript и основы программирования

9. Что такое JavaScript и каковы его основные возможности в веб-разработке?

- JavaScript — мультипарадигменный язык программирования для веб
- Добавляет интерактивность и динамику на веб-страницы
- Позволяет управлять DOM, обрабатывать события, отправлять асинхронные запросы
- Работает как на клиенте (в браузере), так и на сервере (Node.js)

10. Как подключить JavaScript к веб-странице?

- Внешний файл:
- Встроенный скрипт:
- Инлайн-обработчики событий (не рекомендуется): `onclick="function()"`
- Атрибуты `async/defer` для асинхронной загрузки:

11. Что такое DOM и как JavaScript взаимодействует с ним?

- DOM (Document Object Model) — представление HTML-документа в виде дерева объектов
- JavaScript может получать доступ к элементам через методы как `getElementById()`, `querySelector()`
- Может изменять содержимое (`innerText`, `innerHTML`), атрибуты, стили и структуру
- Позволяет добавлять/удалять/перемещать элементы на странице

12. Объясните принцип работы событий в JavaScript.

- События — это действия, происходящие в браузере (клик, ввод текста, загрузка страницы)
- Обработчики событий — функции, выполняющиеся в ответ на событие
- Добавляются через `addEventListener()` или онлайн-атрибуты (`onclick`, `onload`)
- События распространяются через `capturing` (захват) и `bubbling` (всплытие)

Серверные технологии и взаимодействие клиент-сервер

13. Что такое HTTP и как работают HTTP-запросы?

- HTTP (HyperText Transfer Protocol) — протокол передачи данных в Интернете
- Основан на модели запрос-ответ
- Основные методы: GET (получить), POST (отправить), PUT (обновить), DELETE (удалить)
- Включает заголовки, статус-коды и тело сообщения

14. Что такое AJAX и как его использовать?

- AJAX (Asynchronous JavaScript and XML) — технология асинхронного обмена данными с сервером
- Позволяет обновлять части страницы без перезагрузки
- Использует XMLHttpRequest или fetch API
- Современный подход — использование fetch() и Promise

15. Что такое REST API и каковы его основные принципы?

- REST (Representational State Transfer) — архитектурный стиль для веб-сервисов
- Принципы: клиент-сервер, отсутствие состояния, кэширование, единый интерфейс
- Использует стандартные методы HTTP для операций CRUD
- Данные обычно передаются в форматах JSON или XML

16. Опишите основные серверные языки и технологии для веб-разработки.

- PHP: популярный язык для генерации динамического контента
- Node.js: JavaScript на сервере, асинхронная обработка запросов
- Python (Django, Flask): мощные фреймворки для быстрой разработки
- Ruby (Ruby on Rails): элегантный язык с конвенционным фреймворком
- Java (Spring): enterprise-решение для крупных проектов

Базы данных и хранение данных

17. Какие типы баз данных используются в веб-разработке?

- Реляционные (SQL, PostgreSQL): структурированные данные с таблицами и отношениями
- NoSQL (MongoDB, Redis): гибкая схема, подходит для масштабирования
- In-memory: высокопроизводительное хранение в оперативной памяти
- Графовые (Neo4j): для связанных данных и социальных графов

18. Объясните разницу между локальным хранилищем, сессионным хранилищем и cookie.

- LocalStorage: постоянное хранение данных в браузере, не имеет срока действия
- SessionStorage: данные существуют только в рамках одной сессии браузера
- Cookies: небольшие файлы, хранящиеся на клиенте, отправляются с каждым запросом, имеют срок действия

Адаптивный веб-дизайн и мобильная разработка

19. Что такое адаптивный веб-дизайн и почему он важен?

- Подход к созданию сайтов, обеспечивающий оптимальное отображение на разных устройствах
- Использует медиа-запросы, гибкие сетки и изображения
- Улучшает пользовательский опыт и влияет на SEO
- Решает проблему разнообразия устройств и размеров экранов

20. Опишите основные принципы создания адаптивных веб-страниц.

- Использование viewport meta-тега для правильного масштабирования
- Применение относительных единиц измерения (% , em, rem, vw, vh)
- Использование CSS-медиазапросов (@media) для разных размеров экрана
- Гибкие изображения (max-width: 100%, height: auto)
- Mobile-first подход (сначала разработка для мобильных устройств)

Производительность и оптимизация

21. Какие методы оптимизации веб-страниц вы знаете?

- Минификация и сжатие CSS, JavaScript и HTML
- Оптимизация изображений (сжатие, lazy loading)
- Уменьшение количества HTTP-запросов (объединение файлов)
- Использование CDN (Content Delivery Network)
- Кэширование ресурсов на стороне клиента

22. Что такое Critical Rendering Path и как его оптимизировать?

- Последовательность шагов, которые браузер выполняет для отображения страницы
- Включает построение DOM, CSSOM, Render Tree, Layout, Paint
- Оптимизация: размещение CSS вверху страницы, отложенная загрузка JS, уменьшение количества блокирующих ресурсов

Безопасность веб-приложений

23. Какие основные угрозы безопасности существуют для веб-приложений?

- Cross-Site Scripting (XSS): внедрение вредоносного кода
- SQL-инъекции: атака на базу данных через незащищенные входные данные
- Cross-Site Request Forgery (CSRF): выполнение действий от имени пользователя
- Man-in-the-Middle (MitM): перехват коммуникации между клиентом и сервером
- DDoS-атаки: перегрузка сервера большим количеством запросов

24. Какие меры можно предпринять для защиты веб-приложений?

- Валидация и санитизация пользовательского ввода
- Использование параметризованных запросов к базе данных
- Защита от CSRF с помощью токенов
- Внедрение HTTPS для шифрования данных
- Надлежащее управление сессиями и аутентификацией

Современные технологии и фреймворки

25. Какие современные фреймворки и библиотеки для фронтенд-разработки вы знаете?

- React: библиотека для создания пользовательских интерфейсов
- Vue.js: прогрессивный фреймворк для создания UI
- Angular: полноценный фреймворк от Google с TypeScript
- Svelte: компилирует код в эффективный ванильный JavaScript

26. Что такое SPA, MPA и PWA? В чем их отличия?

- SPA (Single Page Application): одностраничное приложение, динамически обновляющее контент
- MPA (Multi Page Application): классическое многостраничное приложение с перезагрузкой страниц
- PWA (Progressive Web App): веб-приложение с возможностями нативного приложения (офлайн-работа, push-уведомления)

27. Что такое WebSockets и для чего они используются?

- Технология, обеспечивающая двунаправленную постоянную связь между браузером и сервером
- Используется для чатов, онлайн-игр, биржевых котировок, уведомлений в реальном времени
- В отличие от HTTP, поддерживает постоянное соединение без необходимости повторных запросов

Системы контроля версий и разработка

28. Что такое Git и как он используется в веб-разработке?

- Распределенная система контроля версий
- Отслеживает изменения в коде и позволяет нескольким разработчикам работать над проектом
- Основные команды: git add, git commit, git push, git pull, git merge
- Сервисы: GitHub, GitLab, Bitbucket

29. Опишите процесс развертывания (депоя) веб-приложения.

- Подготовка сборки (build): минификация, бандлинг, оптимизация ресурсов

- Выбор хостинга/сервера для размещения
- Настройка окружения (веб-сервер, базы данных, переменные окружения)
- Загрузка файлов или настройка CI/CD процесса
- Тестирование после развертывания и мониторинг

30. Что такое NPM и как он используется в веб-разработке?

- Node Package Manager — менеджер пакетов для экосистемы JavaScript
- Используется для установки, управления и публикации JavaScript-библиотек
- package.json — файл для описания проекта и его зависимостей
- Основные команды: npm install, npm start, npm run, npm publish