РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

В.С. Якуничев, М.И. Муконина

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

МДК.03.02 «Фронтенд-разработка (клиентская часть)»

для специальности
09.02.09 Веб-разработка

Ростов-на-Дону
2025

СОДЕРЖАНИЕ

1 САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ

Самостоятельная работа студентов – способ активного, целенаправленного приобретения студентом новых для него знаний и умений без непосредственного участия в этом процесса преподавателей.

Организационные мероприятия, обеспечивающие нормальное функционирование самостоятельной работы студента, должны основываться на следующих предпосылках:

– самостоятельная работа должна быть конкретной по своей предметной направленности;

– самостоятельная работа должна сопровождаться эффективным, непрерывным контролем и оценкой ее результатов.

Результаты самостоятельной подготовки проявляются в активности студента на занятиях и качественном уровне сделанных домашних заданиях и курсовых проектов, и других форм текущего контроля.

2 СТРУКТУРА ДИСЦИПЛИНЫ

Введение. Дисциплина «Фронтенд-разработка (клиентская часть)» реализуется в учебном плане среднего профессионального образования по специальности 09.02.09 «Веб-разработка».

Место дисциплины «Фронтенд-разработка (клиентская часть)» в структуре образовательной программы

Место дисциплины «Фронтенд-разработка (клиентская часть)» в структуре образовательной программы среднего профессионального образования по специальности 09.02.09 «Веб-разработка» определяется как базовый курс по разработке клиентской части веб-приложений.

Дисциплина отнесена к профессиональному циклу Образовательной программы, реализуется в рамках профессионального модуля ПМ.03 «Разработка веб-приложения на стороне клиента».

Дисциплина реализуется в четвертом семестре.

Форма контроля – экзамен.

Целью освоения дисциплины является теоретическая и практическая подготовка студентов в области разработки клиентской части веб-приложений с использованием современного языка программирования JavaScript, объектной модели документа (DOM), а также механизмов асинхронного взаимодействия. Знания, полученные в результате освоения дисциплины, помогут при создании интерактивных пользовательских интерфейсов и

одностраничных приложений (SPA), которые используются в области повсеместно.

Задачи освоения дисциплины состоят в изучении синтаксиса языка JavaScript (стандарт ES6+), принципов функционального программирования, событийной модели браузера, методов манипуляции DOM-деревом, работы с API и локальным хранилищем данных.

В результате освоения дисциплины обучающийся должен:

Знать:

– основные этапы выполнения JavaScript-кода в браузере;

– базовые типы данных, операторы и управляющие конструкции языка;

– принципы работы функций, замыканий и областей видимости;

– структуру Объектной Модели Документа (DOM) и методы навигации по ней;

– событийную модель JavaScript (всплытие, перехват, делегирование);

– методы асинхронного программирования (Callback, Promise, Async/Await);

– технологии хранения данных на стороне клиента (Web Storage).

– Уметь:

– грамотно проектировать архитектуру клиентского приложения;

– создавать интерактивные элементы интерфейса;

– манипулировать содержимым и стилями страницы через JavaScript;

– реализовывать валидацию форм и обработку пользовательского ввода;

– осуществлять сетевые запросы для получения данных без перезагрузки страницы;

– использовать инструменты отладки (Console, Debugger) в браузере.

В результате освоения учебной дисциплины обучающийся должен обладать общими и профессиональными компетенциями, включающими в себя способность:

– ПК-3.1 - Проектировать структуры разделов информационных ресурсов с целью создания эскиза и прототипа интерфейса пользователя.

– ПК-3.2 - Разрабатывать интерфейс пользователя для информационных ресурсов с использованием стандартов в области веб-разработки.

– ПК-3.3 - Создавать структуру кода веб-страницы информационных ресурсов в соответствии с дизайн-макетом.

– ПК-3.4 - Создавать программный код на стороне клиента в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков.

Содержание дисциплины

Семестр № 4

1. Базовые конструкции языка

Роль JavaScript в современной веб-разработке. Среды выполнения (Browser, Node.js). Подключение скриптов: атрибуты async и defer. Инструменты разработчика (Console). Переменные (var, let, const) и их области видимости. Типы данных: примитивы (String, Number, Boolean, Null, Undefined, Symbol, BigInt) и объекты. Операторы и приведение типов.

2. Структуры данных и их обработка

Условные операторы: if, else, тернарный оператор, конструкция switch. Циклы: for, while, do...while. Управление потоком: break, continue. Обработка ошибок: конструкция try...catch. Строгий режим ('use strict') и его особенности. Функции: объявление, параметры, возврат значения, стрелочные функции, замыкания.

3. Динамическая работа со страницей

Объектная модель документа (DOM): структура DOM-дерева, типы узлов. Навигация по дереву (parentNode, children). Поиск элементов: getElementById, querySelector, querySelectorAll. Манипуляции с элементами: создание (createElement), вставка (append, prepend), удаление (remove), клонирование. Работа с атрибутами и классами (classList).

4. Клиентское хранение и асинхронность

Механизмы хранения данных в браузере: Web Storage API (localStorage, sessionStorage) и Cookies. Формат обмена данными JSON (JSON.stringify, JSON.parse). Асинхронность: таймеры (setTimeout, setInterval), модель Event Loop. Взаимодействие с сервером: Fetch API, отправка GET и POST запросов, обработка ответов.

5. Интеграция данных и разработка приложений

Событийная модель: понятие события, способы назначения обработчиков, объект события (event), фазы распространения (всплытие и погружение). Делегирование событий. Работа с формами: доступ к элементам, события (submit, input, change), валидация данных. Принципы создания одностраничных приложений (SPA).

3 РАСПРЕДЕЛЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО РАЗДЕЛАМ

Цель методических рекомендаций: помочь студентам при самостоятельном изучении учебной программы с использованием лекционных материалов и рекомендуемой учебно-методической литературы.

Номер раздела данной дисциплины	**Наименование тем, вопросов, вынесенных для самостоятельного изучения**
Семестр № 4	
Тема 1. Особенности отладки в консоли браузера	Нюансы преобразования типов при сравнении (== vs ===). Ложные значения (Falsy values). Работа с битовыми операторами (обзорно). Особенности использования var, let и const в разных контекстах.
Тема 2. Работа с объектами и массивами	Метки в циклах (Labels). Оптимизация производительности циклов. Паттерны обработки ошибок. Различия между switch и цепочкой if-else.
Тема 3. Оптимизация обработки событий на странице	Оптимизация вставки элементов через DocumentFragment. Теневой DOM (Shadow DOM). Различия «живых» и статических коллекций узлов. Методы getElementsByClassName и getElementsByTagName vs querySelectorAll.
Тема 4. Работа с асинхронным кодом	Политика безопасности при работе с localStorage и Cookies (HttpOnly, Secure). Ограничения памяти браузера. Микрозадачи и макрозадачи в Event Loop. Отмена асинхронных операций с помощью AbortController.
Тема 5. Организация структуры кода в проекте	Пользовательские события (CustomEvent). Паттерны Drag'n'Drop (перетаскивание). Отмена действий браузера (preventDefault). Принципы роутинга в SPA на чистом JavaScript.

4 КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОПРОВЕРКИ

1. Виды и формы самостоятельной работы студентов по дисциплине «Фронтенд-разработка (клиентская часть)»:

– систематическая проработка лекций, учебной и специальной технической литературы;

– выполнение лабораторных заданий;

– оформление отчетов по лабораторным работам и подготовка к их защите;

– подготовка к экзамену.

2. Методические рекомендации для студентов по конкретным видам самостоятельной работы:

– систематическая проработка лекций и литературы;

– оформление отчетов по лабораторным работам.

Методические рекомендации:

– обратитесь к методическим указаниям по проведению лабораторных работ и оформите работу, указав название, цель, порядок проведения, код скриптов и выводы;

– подготовьтесь к защите выполненной работы: повторите теорию и ответьте на контрольные вопросы.

Показатели оценки результатов внеаудиторной самостоятельной работы:

– оформление работ в соответствии с требованиями;

– качественное выполнение этапов программирования;

– понимание алгоритмов работы кода; обоснованность ответов на вопросы.

3. Подготовка к экзамену. Методические рекомендации:

– внимательно изучите конспекты лекций;

– разберитесь с синтаксисом языка и методами API;

– ответьте на контрольные вопросы для самопроверки;

– реализуйте практические примеры кода.

4 Освоив теоретический материал, подготовьтесь к компьютерному тестированию. Примерный перечень вопросов приведен в соответствующих методических рекомендациях.

Показатели оценки результатов внеаудиторной самостоятельной работы:

– качество уровня освоения учебного материала;

– умение использовать теоретические знания при выполнении лабораторных и практических работ или ответе на вопросы;

– обоснованность и четкость изложения ответа.

Представленные в подразделе контрольные вопросы разработаны на основе рабочей программы и включают вопросы по разделам и темам курса. Приводится также ссылки на литературу, которую необходимо изучить при самостоятельной проработке тем.

Перечень информационных технологий

Для проведения лекционных и лабораторных занятий рекомендуется использовать программное обеспечение:

1. Debian, Simply Linux, Microsoft Windows. Системное программное обеспечение
2. LibreOffice. Программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др.
3. Visual Studio Community. Полнофункциональная, расширяемая и бесплатная интегрированная среда разработки для создания современных приложений Android, iOS и Windows, а также веб-приложений и облачных служб
4. Denwer. Набор дистрибутивов (локальный сервер WAMP) и программная оболочка, предназначенные для создания и отладки сайтов (веб-приложений, прочего динамического содержимого интернет-страниц) на локальном ПК (без необходимости подключения к сети Интернет) под управлением ОС Windows.

Контрольные вопросы

1. Каковы основные отличия стандарта ES6 от предыдущих версий JS?
2. Как подключить JavaScript-файл к HTML-странице? В чем разница между async и defer?
3. Перечислите все примитивные типы данных в JavaScript.
4. В чем разница между null и undefined?
5. Объясните различия между объявлением переменных через var, let и const.
6. Как работает приведение типов (type coercion) в условных операторах?
7. Что такое "ложные" (falsy) значения? Перечислите их.
8. В чем разница между операторами == и ===?
9. Опишите синтаксис тернарного оператора.
10. Как работают циклы for...of и for...in? В чем их отличие?
11. Что такое всплытие (hoisting) переменных и функций?
12. Чем отличается Function Declaration от Function Expression?
13. Что такое стрелочные функции и как в них работает ключевое слово this?
14. Объясните понятие "Замыкание" (Closure). Приведите пример.
15. Какие методы существуют для работы с концом и началом массива (push, pop, shift, unshift)?
16. Как работают методы перебора массива map, filter и reduce?

17. Как скопировать объект? В чем разница между поверхностным и глубоким копированием?
18. Что такое деструктуризация (destructuring) объектов и массивов?
19. Что такое DOM? Из каких узлов состоит дерево документа?
20. Как найти элемент на странице по ID, классу и CSS-селектору?
21. В чем разница между HTMLCollection и NodeList?
22. Чем отличаются свойства innerHTML, innerText и textContent?
23. Как изменить CSS-класс элемента через JavaScript? Свойство classList.
24. Как создать новый элемент и вставить его в DOM-дерево?
25. Что такое атрибуты элемента и как с ними работать (getAttribute, setAttribute)?
26. Что такое событие? Как добавить обработчик события (addEventListener)?
27. Объясните фазы распространения события: погружение и всплытие.
28. Что такое объект события (event)? Какие полезные свойства он содержит?
29. Как отменить действие браузера по умолчанию (preventDefault)?
30. Как остановить всплытие события (stopPropagation)?
31. Что такое делегирование событий и зачем оно нужно?
32. Как получить данные из формы? Событие submit.
33. Какие события возникают при вводе текста в поле (input, change, focus, blur)?
34. В чем разница между localStorage и sessionStorage?
35. Как сохранить объект в localStorage? Методы JSON.stringify и JSON.parse.
36. Как работает асинхронность в JavaScript? Что такое Event Loop?
37. Как работают таймеры setTimeout и setInterval?
38. Что такое Promise? Какие состояния есть у промиса?
39. Как использовать цепочку .then() и обработку ошибок .catch()?
40. В чем преимущества синтаксиса async / await?
41. Как сделать HTTP-запрос с помощью fetch?
42. Как обработать ошибки при использовании fetch (сетевые и ошибки статуса)?
43. Что такое JSON? Его структура и отличия от объекта JS.
44. Что такое CORS и почему возникают ошибки кросс-доменных запросов?
45. Как организовать модульную структуру кода (import/export)?

5 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Индивидуальное задание назначается преподавателем при собеседовании со студентом. Логика работы и функционал будущего веб-приложения согласовываются с преподавателем.

Студент самостоятельно или с помощью преподавателя выбирает из предложенного списка тему для своей будущей индивидуальной работы. Тема работы может быть предложена студентом помимо указанного списка, но она обязательно должна быть согласована с преподавателем.

В рамках индивидуальной работы по дисциплине «Фронтенд-разработка (клиентская часть)» студент должен разработать Одностраничное Веб-Приложение (Single Page Application - SPA) для управления данными определенной предметной области, согласно индивидуальному заданию.

Разрабатываемое приложение должно быть написано на чистом JavaScript (ES6+) с использованием HTML5 и CSS3.

В качестве клиентского приложения должен использоваться любой современный браузер. Использование серверных языков (PHP) и баз данных (MySQL) в данном семестре не требуется (все данные должны обрабатываться на клиенте).

Разрабатываемый проект должен соответствовать следующим требованиям:

– **Архитектура SPA:** приложение не должно перезагружать страницу при переходе между разделами или выполнении действий.

– **Хранение данных:** использование **localStorage** (или sessionStorage) для сохранения состояния приложения между сессиями браузера.

– **Динамичность:** интерфейс должен генерироваться через JavaScript (методы document.createElement, innerHTML), статический HTML-код должен быть минимален.

– **Интерактивность:** наличие форм добавления/редактирования данных, кнопок удаления, фильтрации и сортировки списков.

– **Функционал:** реализация полного цикла CRUD (Create, Read, Update, Delete) — создание, чтение, обновление и удаление записей.

Разрабатываемая система должна решать следующие задачи:

– динамическое отображение и обновление данных без перезагрузки;

– валидация вводимых пользователем данных до сохранения;

– удобное управление списками объектов (сортировка, поиск, фильтры).

Примерный перечень предметных областей, на основании которых может быть сформирован индивидуальный вариант задания (тема веб-приложения):

1. Менеджер тарифов оператора связи: список тарифов, возможность добавления новых, редактирование цен, калькулятор стоимости услуг.
2. Планировщик студенческой конференции: формирование расписания, добавление спикеров, отметка «Избранное» для докладов.
3. Погодный дашборд: добавление городов в список, отображение погоды (через mock-данные или API), удаление городов, графики температур.
4. Учет волонтеров депутата: база данных волонтеров (ФИО, контакт), распределение задач, статус выполнения задачи.
5. Персональная электронная библиотека: список книг, статусы (читаю/прочитал), личный рейтинг, фильтрация по жанрам.
6. Управление автостоянкой: интерактивная схема мест (занято/свободно), фиксация времени въезда, расчет стоимости парковки.
7. Статистика баскетбольной лиги: ведение счета матча в реальном времени, протокол игры, список игроков команды.
8. Инвентаризация IT-отдела: учет оборудования, закрепление за сотрудниками, смена статуса (на складе/в ремонте/выдано).
9. Электронный дневник (Калькулятор успеваемости): ввод оценок по предметам, автоматический расчет среднего балла, график успеваемости.
10. График отпусков кафедры: интерактивный календарь, добавление периодов отпуска сотрудников, проверка пересечений дат.
11. Журнал посещаемости: список группы, чекбоксы присутствия на текущую дату, сохранение истории посещений, статистика пропусков.
12. Табло железнодорожного вокзала: список рейсов, фильтрация по направлению, статус рейса (по расписанию/задержан/отменен).
13. Каталог городской библиотеки: поиск книг, учет выданных книг (кому и когда), расчет даты возврата, должники.
14. Запись в медицинскую клинику: выбор врача и временного слота, бронирование времени, просмотр своих записей.
15. Калькулятор копировального центра: конструктор заказа (бумага, цветность, переплет), динамический расчет стоимости тиража.
16. Корзина интернет-магазина обуви: добавление товаров, изменение количества, удаление, подсчет итоговой суммы со скидками.
17. Калькулятор услуг ЖКХ: ввод показаний счетчиков (вода, свет), сохранение истории показаний, расчет суммы к оплате по тарифам.
18. Трекер тренировок: журнал упражнений (подходы/веса), таймер отдыха, статистика прогресса за неделю.
19. Запись в парикмахерский салон: список мастеров, календарь занятости, добавление записи на стрижку, история посещений.
20. Бронирование билетов в театр: интерактивная схема зала, выбор мест кликом, корзина билетов, итоговая стоимость.
21. Калькулятор зарплаты (HR): ввод оклада, часов, премий и налогов, расчет суммы «на руки», формирование расчетного листа.

22. Диспетчер службы доставки: список заказов, смена статусов (готовится/в пути/доставлен), фильтр по курьерам.

23. Трекер фильмов и сериалов: списки «Буду смотреть» и «Просмотрено», личная оценка, поиск по названию.

24. Смета строительной фирмы: добавление работ и материалов, расчет стоимости ремонта, сохранение сметы.

25. Журнал звонков IP-телефонии: список вызовов (входящие/исходящие), длительность, расчет стоимости разговора.

26. Таймер компьютерного клуба: управление игровыми сессиями, старт/стоп времени на ПК, расчет оплаты по тарифу.

27. Конфигуратор заказа гаджетов: выбор модели, цвета и комплектации, динамическое обновление цены и фото.

28. Заявка на Тест-драйв: выбор авто, желаемой даты и времени, проверка доступности слота, сохранение заявки.

29. Склад стройматериалов: таблица остатков, приход/расход товара, уведомление о заканчивающихся позициях.

30. Рейтинг абитуриентов ВУЗа: список поступающих, сортировка по баллам ЕГЭ, фильтрация по специальностям, определение прохождения.

Примерный план выполнения индивидуальной работы

1. Выбор темы. Согласование функционала с преподавателем.
2. Проектирование данных. Описание структуры JSON-объекта, который будет описывать сущность (например, Задача: id, текст, дата, статус).
3. HTML-верстка. Создание каркаса приложения (Header, Main, Footer, контейнеры для списков).
4. Стилизация (CSS). Оформление интерфейса, адаптивность.
5. Разработка Модели (JS). Написание функций для работы с данными (массивом объектов): добавление, поиск, удаление.
6. Разработка Представления (JS). Написание функций отрисовки (render), которые превращают данные в HTML-код.
7. Разработка Контроллера (JS). Назначение обработчиков событий на кнопки, формы, поля ввода.
8. Интеграция с LocalStorage. Добавление сохранения данных при любом изменении.
9. Отладка. Тестирование всех сценариев использования, проверка консоли на наличие ошибок.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература:

1. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для среднего профессионального образования / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Профессиональное образование). — ISBN 978-5-534-16767-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/565693

2. Сысолетин, Е. Г. Разработка интернет-приложений : учебник для среднего профессионального образования / Е. Г. Сысолетин, С. Д. Ростунцев. — Москва : Издательство Юрайт, 2025. — 80 с. — (Профессиональное образование). — ISBN 978-5-534-19603-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/565692

3. Полуэктова, Н. Р. Разработка веб-приложений : учебник для среднего профессионального образования / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-18644-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/567621

Дополнительная литература:

1. **Флэнаган, Д.** JavaScript. Полное руководство : пер. с англ. / Дэвид Флэнаган. – 7-е изд., испр. и доп. – Москва : Вильямс, 2021. – 720 с.

2. **Симпсон, К.** Вы не знаете JS. ES6 и не только / Кайл Симпсон ; пер. с англ. А. А. Слинкина. – 2-е изд. – Санкт-Петербург : Питер, 2022. – 336 с.

3. **Ресиг, Д.** Секреты JavaScript ниндзя / Джон Ресиг, Беэр Бибо. – 2-е изд. – Москва : Вильямс, 2017. – 544 с.

4. **Чинн, Э.** Изучаем React. Переходим на сторону клиентской разработки / Э. Чинн. – Санкт-Петербург : Питер, 2019. – 256 с.

5. **Макфарланд, Д.** JavaScript и jQuery. Исчерпывающее руководство / Дэвид Макфарланд. – 3-е изд. – Москва : Эксмо, 2017. – 880 с.

6. **Крокфорд, Д.** JavaScript: сильные стороны / Дуглас Крокфорд. – Санкт-Петербург : Питер, 2013. – 176 с.

Электронные ресурсы:

1. **Современный учебник JavaScript** [Электронный ресурс]. – Режим доступа: https://learn.javascript.ru/

2. **Документация MDN Web Docs (Раздел JavaScript)** [Электронный ресурс] // Mozilla Developer Network. – Режим доступа: https://developer.mozilla.org/ru/docs/Web/JavaScript

3. **ECMAScript Language Specification** [Электронный ресурс]. – Режим доступа: https://tc39.es/ecma262/