

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

---

М.И. Муконина

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ**  
**СТУДЕНТОВ ПО ДИСЦИПЛИНЕ**

**МДК.03.01 «Основы веб-разработки и верстки»**

для специальности  
09.02.09 Веб-разработка

Ростов-на-Дону  
2025

## СОДЕРЖАНИЕ

1 САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ .....	3
2 СТРУКТУРА ДИСЦИПЛИНЫ.....	3
3 РАСПРЕДЕЛЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО РАЗДЕЛАМ.....	6
4 КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОПРОВЕРКИ.....	8
5 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ.....	11
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	15

## **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 заданий и курсовых проектов, и других форм текущего контроля.

## **2 СТРУКТУРА ДИСЦИПЛИНЫ**

**Введение.** Дисциплина «Основы веб-разработки и верстки» реализуется в учебном плане направления «Веб-разработка».

### **Место дисциплины «Основы веб-разработки и верстки» в структуре образовательной программы**

Место дисциплины «Основы веб-разработки и верстки» в структуре образовательной программы направления «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-приложений на стороне клиента».

Дисциплина реализуется в четвертом семестре.

Форма контроля – экзамен.

Целью освоения дисциплины «Основы веб-разработки и верстки» является формирование у студентов базовых теоретических знаний об архитектуре веб-ресурсов и практических навыков создания (верстки) статических веб-страниц с использованием языка гипертекстовой разметки HTML5 и каскадных таблиц стилей CSS3. Знания, полученные в результате освоения дисциплины, являются фундаментом для дальнейшего изучения клиентского (JavaScript) и серверного (PHP) программирования.

Задачи освоения дисциплины состоят в изучении принципов работы сети Интернет (протоколы HTTP/HTTPS), синтаксиса и семантики HTML, механизмов стилизации элементов через CSS, современных подходов к верстке макетов (Flexbox, Grid Layout), а также принципов адаптивности для отображения сайтов на мобильных устройствах.

В результате освоения дисциплины обучающийся должен:

Знать:

- основные этапы жизненного цикла веб-страницы (от макета до отображения в браузере);
- структуру HTML-документа, семантическое назначение тегов стандарта HTML5;
- синтаксис CSS, типы селекторов, механизмы каскадности и наследования, блочную модель документа;
- современные технологии построения сеток (Flexbox, Grid Layout);
- принципы адаптивной верстки и медиа-запросы;
- требования к кроссбраузерности и валидности кода.

Уметь:

- создавать семантически корректную разметку веб-страниц;
- стилизовать элементы интерфейса, работать с шрифтами, цветами и фонами;
- верстать типовые элементы интерфейса (меню, формы, таблицы, карточки товаров);
- создавать адаптивные макеты, корректно отображающиеся на различных устройствах (ПК, планшеты, смартфоны);
- использовать инструменты разработчика (Browser DevTools) для отладки верстки.

Владеть:

- навыками работы с графическими макетами (Figma/Photoshop) для извлечения параметров верстки;
- инструментами валидации HTML и CSS кода;
- методами оптимизации графического контента для Web;
- базовыми приемами поисковой оптимизации (SEO) на уровне верстки.

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. Основы функционирования веб- технологий и архитектурные принципы интернет- инфраструктуры**

Введение в технологии Интернет. Принцип архитектуры «клиент-сервер». Протоколы передачи гипертекста HTTP/HTTPS. Роль фронтенда, бэкенда и баз данных в создании сайтов. Язык гипертекстовой разметки HTML. Структура веб-документа: декларация `<!DOCTYPE>`, основные разделы `<html>`, `<head>`, `<body>`. Мета-информация страницы. Синтаксис тегов и атрибутов.

#### **2. Интерактивные и структурные элементы HTML**

HTML-элементы для разметки текста: заголовки разных уровней, параграфы, переносы строк. Физическое и логическое форматирование текста. Создание списков: маркированные, нумерованные и списки определений. Вложенные структуры. Гиперссылки: абсолютные и относительные пути, атрибуты навигации. Работа с растровой и векторной графикой на веб-страницах. Создание и оформление таблиц в HTML. Элементы таблицы: строки, ячейки, заголовки. Объединение ячеек и семантическое деление таблицы. HTML-формы для взаимодействия с пользователем. Элементы ввода данных: текстовые поля, кнопки, переключатели, флажки, выпадающие списки. Атрибуты валидации и отправки форм.

#### **3. Семантическое моделирование веб- документов и базовые концепции CSS**

Новые структурные теги стандарта HTML5. Назначение семантической разметки для поисковой оптимизации (SEO) и доступности контента (Accessibility). Отличия блочных и строчных элементов в разметке.

#### **4. Функциональные компоненты CSS**

Каскадные таблицы стилей (CSS). Способы подключения стилей к документу: встроенные, внутренние и внешние стили. Синтаксис CSS-правила. Основные типы селекторов: по тегу, классу, идентификатору, атрибуту. Комбинаторы и псевдоклассы. Механизмы наследования свойств и определение приоритетов (специфичность) стилей. Блочная модель документа (Box Model). Внешние отступы (margin), внутренние поля (padding), границы (border) и контентная область. Расчет ширины и высоты элементов (box-sizing).

Свойства для оформления текста и типографики. Работа с фоном элемента: цвета, изображения, градиенты.

### **5. Позиционирование и современные макеты в CSS**

Схема нормального потока документа. Свойство position и его значения: статическое, относительное, абсолютное и фиксированное позиционирование. «Липкое» (sticky) позиционирование. Контекст наложения и управление перекрытием элементов (z-index). Создание одномерных макетов с помощью технологии Flexbox. Понятие осей, flex-контейнера и flex-элементов. Управление выравниванием и порядком элементов. Технология CSS Grid Layout для создания двумерных сеток. Определение шаблонов строк и колонок, именованные области, единицы измерения fr. Адаптивное размещение элементов в сетке.

## **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.

<b>Номер раздела данной дисциплины</b>	<b>Наименование тем, вопросов, вынесенных для самостоятельного изучения</b>
<b>Семестр № 4</b>	
Тема 1. Эволюция стандартов веб-технологий. Организации по стандартизации (W3C, WHATWG). Различия XHTML и HTML5	Первые браузеры и простейшие версии HTML (HTML 1.0, 2.0). Конкуренция Netscape Navigator и Internet Explorer. Ответная реакция: создание консорциума W3C (1994) и первые попытки стандартизации (HTML 3.2, 4.0). Влияние идеологии XML: синтаксическая строгость, разделение структуры и представления. Развитие XHTML 1.0 и 1.1 как «перезагрузки» HTML на основе XML. Доктрина «Семантического Веба» и Web Standards Project (WaSP). Формирование рабочей группы WHATWG (Apple, Mozilla, Opera) и создание спецификации Web Applications 1.0 (позже HTML5). История создания, миссия и принципы: «Единый веб для всех». Организационная структура: члены, рабочие группы, процесс рекомендаций (W3C Recommendation). Модель «живого стандарта» (HTML Living

	Standard): постоянное обновление без версий. Синергия: WHATWG как двигатель инноваций, W3C как площадка для фиксации стабильных версий и развития смежных стандартов.
Тема 2. Стандарты доступности веб-интерфейсов (WCAG). Использование ARIA-атрибутов для улучшения доступности форм	Введение в веб-доступность. Вспомогательные технологии (AT): программы чтения с экрана (NVDA, JAWS, VoiceOver), скрин-ридеры браузеров, голосовое управление, специальные клавиатуры и др. Обзор WCAG (Web Content Accessibility Guidelines). Версии и эволюция: WCAG 1.0, 2.0, 2.1, 2.2, будущее WCAG 3.0 (Silver). Четыре принципа доступности POUR (основа WCAG). Проверка доступности: автоматические инструменты (axe, WAVE, Lighthouse), ручное тестирование с клавиатуры и скринридером, вовлечение пользователей с ограниченными возможностями. Использование ARIA-атрибутов для улучшения доступности форм. ARIA для описания состояния и свойств полей ввода. ARIA для кастомных виджетов в формах. Паттерны и практические примеры.
Тема 3. Понятие «CSS Reset» и «Normalize». Расчет веса (специфичности) селекторов и решение конфликтов стилей	Проблема кроссбраузерной несогласованности. Различия в User Agent Stylesheet (стилях браузера по умолчанию) для элементов. Подход CSS Reset: «Жесткий сброс». Классический пример Эрика Мейера (Eric Meyer's Reset CSS). Подход Normalize.css: «Умная нормализация». Принципы работы Normalize.css. Расчет веса (специфичности) селекторов. Визуальная модель (концепция "веса"): Очки за инлайн-стили, ID, классы, элементы. Методология БЭМ (Блок-Элемент-Модификатор). CSS-модули / Scoped CSS: Автоматическая генерация уникальных имен классов на этапе сборки. Работа с CSS-фреймворками (Bootstrap, Tailwind).

Тема 4. Введение в CSS-препроцессоры (Sass/SCSS, Less): переменные, вложенность, миксины	Концепция препроцессора. Настройка рабочего окружения. Основные возможности препроцессоров. Вложенность (Nesting). Миксины (Mixins). Простые миксины (без аргументов). Миксины с аргументами (параметрами). Карты (Maps) и Списки (Lists). Импорт и модульность.
Тема 5. CSS-анимации и переходы (transition). Оптимизация верстки для мобильных устройств (Mobile First)	CSS-переходы (Transitions). Продвинутое функции плавности (Timing Functions). CSS-анимации (Animations). Трансформации (Transforms) – основа анимаций. 2D-трансформации. 3D-трансформации (перспектива). Производительность анимаций. Философия и принцип Mobile First. Техническая реализация Mobile First в CSS. Адаптивная типографика. Адаптивные изображения и медиа.

#### 4 КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОПРОВЕРКИ

1. Виды и формы самостоятельной работы студентов по дисциплине «Основы веб-разработки и верстки»:

- 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. Приводятся 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. В чём заключается суть архитектуры «клиент сервер»?
2. Чем отличаются протоколы HTTP и HTTPS? Какова роль фронтенда в веб разработке?
3. В чём состоит задача бэкенда?
4. Для чего нужны базы данных в веб приложениях?
5. Что обозначает декларация `<!DOCTYPE>` в HTML?
6. Перечислите и кратко опишите три основных раздела HTML документа: `<html>`, `<head>`, `<body>`.
7. Какие виды мета информации можно разместить в секции `<head>`?
8. Что такое HTML тег?
9. Для чего используются атрибуты в HTML тегах? Приведите пример тега с атрибутом.
10. Какие HTML элементы используются для разметки заголовков?
11. Как создать параграф текста в HTML? Какой тег для этого применяется?
12. В чём разница между физическим и логическим форматированием текста в HTML? Приведите по одному примеру тега для каждого типа.
13. Как создать маркированный список в HTML? Какой тег используется для его контейнера?
14. Чем отличается нумерованный список от списка определений?
15. Что такое вложенные структуры в HTML? Приведите пример вложенного списка.
16. В чём разница между абсолютными и относительными путями в гиперссылках? Приведите примеры обоих.
17. Какие атрибуты используются для настройки навигации в гиперссылках (например, открытие в новой вкладке)?
18. Как вставить растровое изображение в HTML документ? Какой тег и атрибуты для этого нужны?
19. Какие элементы используются для создания таблиц в HTML? Перечислите основные теги для строк, ячеек и заголовков.
20. Перечислите 3–4 новых структурных тега, появившихся в HTML5. В чём их преимущество перед универсальными `<div>`?
21. Почему семантическая разметка важна для SEO?
22. Как семантическая разметка улучшает доступность контента (Accessibility)?
23. В чём ключевое отличие блочных элементов от строчных в HTML?
24. Почему важно использовать семантические теги?
25. Перечислите три способа подключения CSS к HTML документу. В чём преимущества и недостатки каждого?
26. Каков синтаксис CSS правила? Приведите пример правила для изменения цвета текста параграфа.
27. Какие типы селекторов существуют в CSS? Приведите примеры селекторов по тегу, классу, идентификатору и атрибуту.

28. Что такое комбинаторы в CSS? Приведите пример использования комбинатора потомков.
29. Опишите блочную модель документа (Box Model). Перечислите четыре основных компонента и их роль в расчёте размеров элемента.

## 5 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Индивидуальное задание назначается преподавателем при собеседовании со студентом. Структура и дизайн будущего сайта 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 «Основы веб-разработки и верстки» студент должен разработать статический многостраничный веб-сайт (фронтенд-часть) для представления информации определенной предметной области, согласно индивидуальному заданию.

Разрабатываемый сайт должен быть построен на основе 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 (сайт должен состоять из статических .html файлов).

Разрабатываемый проект должен соответствовать следующим требованиям:

1. **Многостраничность:** сайт должен содержать минимум 3-4 связанные страницы (Главная, О нас/Контакты, Каталог/Услуги, Детальная страница).
2. **Адаптивность:** интерфейс должен корректно отображаться на десктопах, планшетах и мобильных телефонах (использование @media запросов).
3. **Технологии:** использование Flexbox и/или CSS Grid Layout для построения сеток.
4. **Семантика:** использование семантических тегов HTML5 (<header>, <nav>, <main>, <footer> и др.).
5. **Контент:** наличие таблиц, списков, изображений и форм (визуальная часть формы без отправки на сервер).

Разрабатываемая система должна решать следующие задачи визуализации:

- представление текстовой и графической информации;
- удобная навигация по разделам сайта;
- адаптация контента под размер экрана пользователя.

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

1. Сайт оператора связи: тарифы, услуги, карта покрытия, форма заявки на подключение.
2. Сайт студенческой конференции: программа мероприятий, список спикеров (ФИО, регалии), секции, форма регистрации.
3. Погодный портал: верстка дашборда с прогнозом погоды (таблицы температур, иконки осадков, графики ветра).
4. Сайт кандидата в депутаты: биография, предвыборная программа, новости, форма для волонтеров.
5. Персональная электронная библиотека: каталог книг (обложка, автор, жанр, описание), страница детального просмотра книги.
6. Сайт автостоянки: схема парковочных мест, тарифы по времени, правила пользования, контакты.
7. Сайт баскетбольной лиги: турнирная таблица, карточки команд (логотип, тренер, город), статистика игр.
8. Информационный портал IT-отдела: каталог компьютерного оборудования (спецификации, фото), инструкции для сотрудников.
9. Интерфейс электронного дневника: таблицы с оценками по предметам, расписание экзаменов, профиль студента.
10. Корпоративный портал кафедры: график отпусков (визуализация в виде таблицы/календаря), список сотрудников, контакты.
11. Интерфейс журнала преподавателя: таблица посещаемости и успеваемости, список студентов группы.
12. Сайт железнодорожного вокзала: онлайн-табло расписания поездов (номер, маршрут, время), информация об услугах вокзала.
13. Сайт городской библиотеки: правила для читателей, поиск по каталогу (верстка результатов поиска), режим работы.
14. Сайт медицинской клиники: расписание врачей, прайс-лист услуг, страница «Оформление больничного».
15. Сайт копировального центра: перечень услуг (копирование, печать), калькулятор цен (визуальная верстка), контакты.
16. Интернет-каталог обуви: витрина товаров (мужская, женская, детская), карточка товара с выбором размера и цены.
17. Сайт управляющей компании (ЖКХ): информация для жильцов, тарифы на услуги, отчетность, контакты мастеров.
18. Сайт спортивного клуба: виды спорта, расписание соревнований, информация о членских взносах, форма вступления.
19. Сайт парикмахерского салона: портфолио мастеров, прейскурант услуг, форма онлайн-записи.
20. Сайт театра: афиша спектаклей, схема зала с ценами билетов, репертуар, труппа театра.

21. Корпоративный сайт (Отдел кадров): страница вакансий, информация о компании, калькулятор зарплаты (верстка интерфейса).
22. Сайт службы доставки: каталог продуктов, зона доставки (карта), условия и акции.
23. Сайт видеопроката / Стриминга: каталог фильмов по жанрам, страница фильма (постер, описание, трейлер).
24. Сайт строительной фирмы: портфолио объектов, прайс-лист на отделочные работы, калькулятор стоимости.
25. Сайт провайдера IP-телефонии: тарифы на междугороднюю связь, личный кабинет пользователя (верстка), поддержка.
26. Сайт компьютерного клуба: список игровых мест (конфигурации ПК), тарифы (день/ночь), барное меню.
27. Интернет-магазин гаджетов: каталог товаров, корзина (верстка таблицы товаров), страница оформления заказа.
28. Сайт автосалона: каталог автомобилей (марки, характеристики, цены), страница «Тест-драйв».
29. Сайт базы стройматериалов: каталог товаров, условия оптовой продажи, схема проезда.
30. Сайт приемной комиссии ВУЗа: информация для абитуриентов, проходные баллы, список специальностей, форма подачи документов.

### **Примерный план выполнения индивидуальной работы**

1. Выбор темы индивидуальной работы. Согласование с преподавателем предметной области и структуры будущего сайта.
2. Проектирование интерфейса и дизайна. Создание эскиза (прототипа) страниц на бумаге или в графическом редакторе (Figma). Определение цветовой схемы и шрифтовых пар.
3. Анализ структуры макета. Выделение основных смысловых блоков страницы (шапка, навигация, основной контент, подвал). Выбор стратегии верстки: использование Flexbox для линейных структур и CSS Grid для основных сеток.
4. Создание HTML-каркаса. Написание разметки с использованием семантических тегов стандарта HTML5 (<header>, <nav>, <main>, <section>, <article>, <aside>, <footer>). Структурирование файлов проекта (создание папок css, img, fonts).
5. Создание и подключение стилей. Создание файла style.css и его подключение к HTML-документу. Последовательное написание стилей: сброс браузерных стилей (reset/normalize), оформление типографики, цветов и отступов.

6. Оформление навигации. Верстка меню, содержащего ссылки на все разделы сайта. Реализация состояний ссылок при наведении (:hover) и активации.

7. Наполнение контентом. Заполнение страниц текстовой информацией и изображениями в соответствии с выбранной тематикой. Оптимизация размера изображений для Веб.

8. Реализация адаптивности. Написание медиа-запросов (@media) в CSS для корректного отображения сайта на мобильных телефонах и планшетах (изменение сетки, размеров шрифтов, скрытие второстепенных блоков).

9. SEO-оптимизация. Для каждой страницы прописать уникальный заголовок (<title>) и мета-теги (description, keywords), соответствующие содержанию страницы.

10. Финальная отладка. Проверка валидности кода (через W3C Validator) и тестирование работы сайта в различных браузерах.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

### Основная литература:

1. **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.
2. **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
3. **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.

### Дополнительная литература:

1. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – 7-е изд. – Санкт-Петербург : Питер, 2025. – 832 с. – (Серия «Бестселлеры O'Reilly»).
2. **Роббинс, Дж.** HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Дженнифер Роббинс; [пер. с англ. М. А. Райтман]. – 4-е изд. – Москва : Эксмо, 2014. – 528 с.
3. **Мейер, Эрик А.** CSS. Карманный справочник / Эрик А. Мейер ; [пер. с англ.]. – 5-е изд. – Москва : ООО «И.Д. Вильямс», 2019. – 208 с.
4. **Фримен, Э.** Изучаем HTML, XHTML и CSS / Эрик Фримен, Элизабет Фримен. – 2-е изд. – Санкт-Петербург : Питер, 2024. – 720 с.

### Электронные ресурсы:

1. **Что такое WCAG** [Электронный ресурс] // Vuejs.org. – Режим доступа: <https://ru.vuejs.org/guide/best-practices/accessibility.html>
2. **Две мощные техники: CSS Reset и normalize.css** [Электронный ресурс] // dev-notes.ru. – Режим доступа: <https://www.dev-notes.ru/articles/css/two-powerful-techniques-css-resetting-and-normalizing/>
3. **Специфичность селекторов в CSS** [Электронный ресурс] // htmlacademy.ru – Режим доступа: <https://htmlacademy.ru/blog/css/which-selector>
4. **CSS-анимации: полное руководство с примерами и кодом** [Электронный ресурс] // Skillbox.ru. – Режим доступа: <https://skillbox.ru/media/code/css-animacii-polnoe-rukovodstvo-s-primerami-i-kodom/>

5. **CSS-препроцессоры** [Электронный ресурс] // Sky.pro. – Режим доступа: <https://sky.pro/wiki/media/kak-rabotat-s-preprocessorami-css-sass-less-stylus/>