

**РОСЖЕЛДОР**

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

---

**УТВЕРЖДАЮ**  
Первый проректор:  
М.А. Кравченко

Кафедра "Вычислительная техника и автоматизированные системы управления"

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ (МЕЖДИСЦИПЛИНАРНОМУ КУРСУ)**

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

**по Учебному плану**

специальности среднего профессионального образования  
09.02.09 Веб-разработка

Квалификация специалиста среднего звена "Разработчик веб-приложений"

Ростов-на-Дону  
2025

## **Содержание**

1. Результаты обучения дисциплины (модуля) .....	3
2. Компетенции, формируемые в процессе изучения дисциплины (модуля) .....	4
3. Оценочные средства для оценки успеваемости студентов .....	5
4. Шкала оценивания в зависимости от уровня сформированности компетенций .....	6

## 1. Результаты обучения дисциплины (модуля)

В результате освоения дисциплины (модуля) обучающийся должен овладеть следующими результатами:

Код и наименование компетенции выпускника	Формулировка требований к степени сформированности компетенции
ПК 3.1 Проектировать структуры разделов информационных ресурсов с целью создания эскиза и прототипа интерфейса пользователя	<p><b>Умения:</b></p> <p>Анализировать типовые разделы сайта (хедер, навигация, основной контент, футер) для создания структуры. Создавать простые эскизы (wireframes) будущих страниц, отражающие расположение основных семантических блоков (header, nav, main, section, footer). Разрабатывать статичный HTML-прототип, используя семантические теги HTML5, для демонстрации структуры контента.</p> <p><b>Знания:</b></p> <p>Основные принципы архитектуры веб-сайтов и клиент-серверного взаимодействия. Назначение и правила применения семантических тегов HTML5 для структурной разметки. Базовые принципы организации навигации с помощью гиперссылок.</p>
ПК 3.2 Разрабатывать интерфейс пользователя для информационных ресурсов с использованием стандартов в области веб-разработки	<p><b>Умения:</b></p> <p>Создавать HTML-документы, соответствующие стандартам синтаксиса и структуры (&lt;!DOCTYPE&gt;, &lt;html&gt;, &lt;head&gt;, &lt;body&gt;, мета-теги). Применять HTML для разметки текстового контента, списков, таблиц, изображений и ссылок в соответствии с их логическим назначением. Создавать HTML-формы с базовыми элементами ввода данных (&lt;input&gt;, &lt;textarea&gt;, &lt;select&gt;). Подключать CSS к HTML-документу и использовать базовые свойства для оформления.</p> <p><b>Знания:</b></p> <p>Синтаксис и структура HTML-документа, назначение основных тегов и атрибутов. Принципы семантической разметки и ее значение для SEO и доступности. Способы подключения CSS, базовый синтаксис правил (селектор, свойство, значение) и работа с основными свойствами оформления. Основы модели визуального форматирования: блочные и строчные элементы, блочная модель (box model).</p>
	<p><b>Умения:</b></p> <p>Анализировать визуальный макет и переводить его в HTML-структуру с</p>

ПК 3.3 Создавать структуру кода веб-страницы информационных ресурсов в соответствии с дизайн-макетом	использованием семантических и базовых тегов. Применять CSS для позиционирования элементов на странице с использованием схемы нормального потока, position (relative, absolute, fixed). Строить одномерные макеты на основе технологии Flexbox. Строить сложные двумерные макеты на основе технологии CSS Grid Layout. Контролировать отступы и размеры элементов с учетом блочной модели CSS (box-sizing, margin, padding).
	<b>Знания:</b> Принципы позиционирования элементов (static, relative, absolute, fixed, sticky) и их влияние на поток документа. Технологию Flexbox: свойства контейнера и элементов для управления выравниванием, порядком и распределением пространства. Технологию CSS Grid: создание сеток, управление размерами колонок/строк (fr, repeat()), размещение элементов по линиям и областям. Правила расчета специфичности CSS-селекторов и каскада для управления применением стилей.
ПК 3.4 Создавать программный код на стороне клиента в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков	<b>Умения:</b> Добавлять простую интерактивность на страницу с помощью CSS (псевдоклассы :hover, :focus). Создавать адаптивные интерфейсы, используя медиазапросы, Flexbox и Grid.
	<b>Знания:</b> Принципы создания адаптивных макетов (mobile-first, breakpoints) с помощью медиазапросов. Способы обработки событий в браузере (например, клик, наведение) и способах реакции на них через CSS.

## 2. Компетенции, формируемые в процессе изучения дисциплины (модуля)

Индекс и Наименование компетенции	Признаки проявления компетенции в соответствии с уровнем формирования в процессе освоения дисциплины
ПК 3.1 Проектировать структуры разделов информационных ресурсов с целью создания эскиза и прототипа интерфейса пользователя	<b>недостаточный</b> уровень: Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы.
ПК 3.2 Разрабатывать интерфейс пользователя для информационных ресурсов с использованием стандартов в области веб-разработки	<b>пороговый</b> уровень: Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.
ПК 3.3 Создавать структуру кода веб-страницы информационных	<b>продвинутый</b> уровень: Компетенции сформированы. Знания обширные, системные. Умения носят репродуктивный характер, применяются к

<p>ресурсов в соответствии с дизайном-макетом</p> <p>ПК 3.4 Создавать программный код на стороне клиента в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков</p>	<p>решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.</p> <p><b>высокий уровень:</b> Компетенции сформированы. Знания твердые, аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка.</p>
--	---

### 3. Оценочные средства для оценки успеваемости студентов

#### **!\*ВВ**

**Язык, используемый для разметки структуры веб-страницы–**

!\*-

JavaScript

!\*+

HTML

!\*-

CSS

!\*-

PHP

#### **!\*ВВ**

**Пара тегов, используемых для обозначения начала и конца веб-страницы в HTML**

– ...

!\*-

<head> и </head>

!\*-

<body> и </body>

!\*+

<html> и </html>

!\*-

<title> и </title>

#### **!\*ВВ**

**Язык CSS предназначен для ...**

!\*-

программирования интерактивных элементов.

!\*-

структурирования содержимого страницы.

!\*+

задания визуального оформления (стилей) элементов.

!\*-

обработки данных на сервере.

**!\*BB**

**Тег, используемый для вставки изображения на страницу – ...**

!\*-

<link>

!\*+

<img>

!\*-

<picture>

!\*-

<figure>

**!\*BB**

**Для создания ссылки на веб-странице, необходимо использовать код ...**

!\*-

<link href="url">Текст</link>

!\*+

<a href="url">Текст</a>

!\*-

<url href="url">Текст</url>

!\*-

<href="url">Текст</href>

**!\*BB**

**Элемент по идентификатору в CSS выбирает селектор ...**

!\*-

.idname

!\*+

#idname

!\*-

idname

!\*-

[idname]

**!\*BB**

**Свойство margin в CSS задаёт ...**

!\*-

внутренний отступ элемента

!\*+

внешний отступ элемента

!\*-

границу элемента

!\*-

ширину элемента

**!\*BB**

**Тег ... используется для создания ненумерованного списка.**

!\*-

<ol>

!\*+

<ul>

!\*-

<li>

!\*-

<dl>

**!\*BB**

**Аббревиатура HTML расшифровывается как ...**

!\*+

Hyper Text Markup Language

!\*-

High Text Marking Language

!\*-

Hyper Text Management Language

!\*-

Hyper Tool Markup Language

**!\*BB**

**Тег ... задаёт заголовок страницы, отображаемый в браузере.**

!\*-

<h1>

!\*+

<title>

!\*-

<header>

!\*-

<head>

**!\*BB**

**Свойство CSS ... задаёт фоновый цвет элемента.**

!\*\_-

color

!\*\_+

background-color

!\*\_-

border-color

!\*\_-

fill-color

**!\*BB**

**Свойство display: flex в CSS ...**

!\*\_-

превращает элемент в сетку (grid)

!\*\_+

превращает элемент в гибкий контейнер (flex container)

!\*\_-

скрывает элемент

!\*\_-

задаёт абсолютное позиционирование

**!\*BB**

**Атрибут ... устанавливает альтернативный текст для изображения.**

!\*\_-

title

!\*\_+

alt

!\*\_-

src

!\*\_-

caption

**!\*BB**

**Тег ... используется для создания поля для ввода данных.**

!\*\_+

<input>

!\*\_-

<form>

```
!*-  
<textarea>  
!*-  
<button>
```

#### **Перечень вопросов для устного опроса:**

- 1) Базовая структура HTML-документа.
- 2) Семантические теги HTML5.
- 3) Формы в HTML: основные элементы, атрибуты.
- 4) Встраивание мультимедиа.
- 5) Способы подключения CSS к HTML.
- 6) Синтаксис CSS: селекторы, свойства, значения, комментарии.
- 7) Блочная модель: margin, border, padding, content (схемы расчёта).
- 8) Системы единиц измерения: пиксели, проценты, em, rem, vh/vw.
- 9) Поточные элементы vs блочные: различия и примеры.
- 10) Плавающие элементы (float): принципы работы и очистка потока.
- 11) CSS Grid: создание гридов.
- 12) DevTools браузера: основные вкладки.

#### **Перечень вопросов для самоподготовки:**

- 1) Историческая эволюция веб-стандартов: от HTML 1.0 до HTML5.
- 2) WHATWG: причины создания, принципы работы, взаимоотношения с W3C.
- 3) XHTML vs HTML5: сравнительный анализ синтаксиса и правил разметки.
- 4) Обработка ошибок в XHTML и HTML5: различия в поведении браузеров.
- 5) Поддержка мультимедиа в HTML5.
- 6) Валидация документов: инструменты и подходы для XHTML и HTML5.
- 7) WCAG: структура стандарта (принципы POUR), уровни соответствия (A, AA, AAA).
- 8) ARIA-атрибуты: базовые концепции, синтаксис, область применения.
- 9) Ключевые ARIA-роли для интерактивных элементов (формы, меню, диалоги).
- 10) Улучшение доступности сложных компонентов: карусели, табы, модальные окна.
- 11) Типичные ошибки в доступности форм и их исправление через ARIA.
- 12) CSS Reset: философия, примеры реализаций, плюсы и минусы.
- 13) Normalize.css: отличия от Reset, преимущества для современных проектов.
- 14) Конфликты стилей: алгоритмы разрешения в браузерах.
- 15) Практические примеры расчёта специфичности: разбор кейсов.
- 16) Каскадирование и наследование: как браузер определяет финальные стили.
- 17) Sass vs SCSS: синтаксические различия, совместимость, выбор формата.
- 18) Вложенность селекторов: преимущества и риски «глубокой» вложенности.
- 19) Функции и операторы: математические вычисления, строковые операции.
- 20) Импорт и модульность: организация кода через частичные файлы.
- 21) CSS-переходы (transition): свойства, длительность, задержка, функции времени.
- 22) Производительность анимаций: свойства will-change, transform, opacity.
- 23) Оптимизация изображений: форматы (WebP, AVIF), адаптивные картинки.
- 24) Шрифты для мобильных: подгрузка, форматы, fallback-стратегии.
- 25) Критические CSS: онлайн-стили, асинхронная загрузка, Critical Path.

- 26) Оптимизация рендеринга: избегание «перерисовки» (repaint) и «перекомпоновки» (reflow).

**Перечень контрольных вопросов к экзамену:**

**Знать:**

- 1) Принцип клиент-серверного взаимодействия в веб-разработке.
- 2) Основные функции (роли) фронтенда, бэкенда и базы данных при создании веб-сайта.
- 3) Протоколы HTTP и HTTPS и их различия.
- 4) Базовые теги структуры HTML-документа и их назначение.
- 5) Правила использования тегов заголовков.
- 6) Разницу между физическим и логическим форматированием текста.
- 7) Обязательные атрибуты тега гиперссылки и тега изображения.
- 8) Типы HTML-списков (неупорядоченный, упорядоченный, список определений) и теги для их создания.
- 9) Структуру HTML-таблицы.
- 10) Основные теги для создания формы и её элементов.
- 11) Атрибуты action и method у тега <form> для отправки данных и метод HTTP-запроса.
- 12) Основные семантические теги HTML5 и их назначение.
- 13) Три основных способа подключения CSS к HTML-документу.
- 14) Основные типы CSS-селекторов.
- 15) Каскад, наследование и специфичность – ключевые механизмы CSS.
- 16) Разницу между блочными и строчными элементами.
- 17) Компоненты блочной модели CSS.
  
- 18) Основные CSS-свойства для управления шрифтом и оформлением текста.
- 19) Свойства для работы с фоном и синтаксис задания градиента.
- 20) Разницу между relative, absolute и fixed позиционированием.
- 21) Назначение свойств flex-контейнера: flex-direction, justify-content, align-items, flex-wrap.
- 22) CSS Grid Layout.
- 23) Свойства для определения структуры сетки: grid-template-columns, grid-template-rows.
- 24) Свойства grid-column и grid-row.
- 25) Свойство gap в Grid.

**Уметь:**

- 1) Создавать корректный HTML-документ.
- 2) Использовать текстовые элементы и теги для логического и физического форматирования для разметки контента.
- 3) Создавать гиперссылки на внешние ресурсы и внутренние разделы сайта.
- 4) Встраивать изображения с обязательным указанием атрибута alt.
- 5) Создавать упорядоченные, неупорядоченные списки и списки определений.
- 6) Строить таблицы данных с заголовочными ячейками, объединением ячеек (и семантическими секциями).
- 7) Разрабатывать HTML-формы с использованием различных полей ввода, их группировкой и подписями.
- 8) Применять семантические теги HTML5 для разметки макета типовой веб-страницы.
- 9) Подключать внешние таблицы стилей (CSS) к HTML-документу и применять стили через селекторы по тегу, классу и идентификатору.
- 10) Использовать различные типы CSS-селекторов и комбинаторов для точечного применения стилей.
- 11) Управлять блочной моделью: задавать размеры, внутренние и внешние отступы, рамки для создания визуальных блоков.

- 12) Применять CSS-свойства для оформления текста и фона элементов.
- 13) Использовать позиционирование для создания сложных интерфейсных элементов: выпадающих меню, фиксированных панелей, модальных окон.
- 14) Создавать гибкие одномерные макеты с помощью Flexbox: выравнивать и распределять элементы внутри контейнера, управлять переносом и порядком.
- 15) Управлять размерами и поведением отдельных flex-элементов с помощью свойств flex-grow, flex-shrink, flex-basis.
- 16) Создавать сложные двумерные макеты с помощью CSS Grid: определять сетку и размещать в ней элементы.
- 17) Использовать единицу измерения fr и функцию repeat() для создания адаптивных сеток.
- 18) Адаптировать макет под различные размеры экранов с использованием медиазапросов.
- 19) Менять расположение, размеры и видимость элементов для мобильных устройств и планшетов.
- 20) Реализовывать адаптивную навигацию с использованием только HTML и CSS.
- 21) Использовать CSS-фреймворк (Bootstrap) на локальных файлах для быстрого построения адаптивных сеток и использования готовых компонентов.
- 22) Анализировать готовый графический макет и декомпозировать его на структурные HTML-элементы и блоки.
- 23) Создавать семантически правильную HTML-разметку для всех страниц проекта, используя методологию именования классов.
- 24) Реализовывать детальную стилизацию всех компонентов макета (кнопки, карточки, формы, типографика) в соответствии с дизайном.
- 25) Обеспечивать кроссбраузерность и валидность верстки.
- 26) Проводить финальное тестирование проекта на разных устройствах и разрешениях экрана.

#### **4. Шкала оценивания в зависимости от уровня сформированности компетенций**

##### ***Описание шкал оценивания компетенций***

<b>Значение оценки</b>	<b>Уровень освоения компетенции</b>	<b>Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)</b>	<b>Шкала оценивания (процент верных при проведении тестирования)</b>
------------------------	-------------------------------------	---	--

<b>Значение оценки</b>	<b>Уровень освоения компетенции</b>	<b>Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образования)</b>	<b>Шкала оценивания (процент верных при проведении тестирования)</b>
Балльная оценка - "удовлетворительно".	Пороговый	Оценка «удовлетворительно» выставляется обучающемуся, который имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения последовательности изложения программного материала и испытывает трудности в выполнении практических навыков.	От 40% до 59%
Балльная оценка - "хорошо".	Базовый	Оценка «хорошо» выставляется обучающемуся, твердо знающему программный материал, грамотно и по существу его излагающему, который не допускает существенных неточностей в ответе, правильно применяет теоретические положения при решении практических работ и задач, владеет необходимыми навыками и приемами их выполнения.	От 60% до 84%
Балльная оценка - "отлично".	Высокий	Оценка «отлично» выставляется обучающемуся, глубоко иочно усвоившему программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагающему, в ответе которого тесно увязываются теория с практикой. При этом обучающийся не затрудняется с ответом при видоизменении задания, показывает знакомство с литературой, правильно обосновывает ответ, владеет разносторонними навыками и приемами практического	От 85% до 100%

Значение оценки	Уровень освоения компетенции	Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образования)	Шкала оценивания (процент верных при проведении тестирования)
		выполнения практических работ.	
Дуальная оценка - "зачтено".	Пороговый, Базовый, Высокий	Оценка «зачтено» выставляется обучающемуся, который имеет знания, умения и навыки, не ниже знания только основного материала, может не освоить его детали, допускать неточности, недостаточно правильные формулировки, нарушения последовательности изложения программного материала и испытывает трудности в выполнении практических навыков.	От 40% до 100%
Балльная оценка - "неудовлетворительно", Дуальная оценка - "не зачтено".	Не достигнут	Оценка «неудовлетворительно, не зачтено» выставляется обучающемуся, который не знает значительной части программного материала, допускает ошибки, неуверенно выполняет или не выполняет практические работы.	От 0% до 39%