


Федеральное государственное образовательное бюджетное  
учреждение высшего образования  
«Финансовый университет при Правительстве Российской Федерации»  
(Финансовый университет)

Красноярский филиал Финуниверситета

УТВЕРЖДАЮ

Заместитель директора  
по учебно – методической работе  
Красноярского филиала  
Финуниверситета

 О.С. Вергейчик  
«»  2026 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по учебному предмету / дисциплине

**ОП.08 ОСНОВЫ ВЕБ-РАЗРАБОТКИ**

(код, наименование)

**09.02.09 Веб-разработка**

(код, наименование специальности)

г. Красноярск – 2026

Фонд оценочных средств по дисциплине ОП.08 Основы веб-разработки разработан на основании федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.09 Веб-разработка

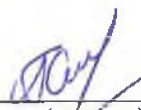
Составители:

Цирулькевич Алена Викторовна, преподаватель

Фонд оценочных средств по дисциплине рассмотрен и рекомендован к утверждению на заседании предметной (цикловой) комиссии общепрофессиональных дисциплин.

Протокол от «02» сентября 2026 г. № 8

Председатель предметной (цикловой)  
комиссии

  
(подпись)

О.А. Полтавец  
(инициалы, фамилия)

1. Паспорт фонда оценочных средств  
по дисциплине ОП.08 Основы веб-разработки  
(код, наименование)

09.02.09 Веб-разработка

(код, наименование специальности)

Результаты обучения (знания, умения)	Общие и профессиональные компетенции	Наименование темы	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
<b>Знать:</b> - архитектуру и принципы функционирования веб-приложений; - основные протоколы интернета (HTTP/HTTPS, FTP); - языки разметки HTML и стилей CSS, их синтаксис и возможности; - основы клиентского программирования на JavaScript; - принципы адаптивной и кросс-браузерной верстки; - системы контроля версий (Git); - основы UI/UX-дизайна; - требования к оптимизации веб-страниц и основы SEO.	ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 3.2	Тема 1.1. Введение в веб-разработку	– Защита практических работ; – Выполнение индивидуальных заданий;	Экзамен
	ОК 02, ОК 09 ПК 3.2	Тема 1.2. Системы контроля версий	– Тестирование;	
	ОК 01, ОК 02 ПК 1.2	Тема 2.1. Язык гипертекстовой разметки HTML	– Экспертное наблюдение за выполнением заданий;	
	ОК 02, ОК 04 ПК 1.2	Тема 2.2. Каскадные таблицы стилей CSS	– Просмотр и защита итогового проекта.	
	ОК 02, ОК 05 ПК 3.2	Тема 2.3. Основы клиентского программирования JavaScript		
	ОК 04, ОК 05 ПК 1.2	Тема 3.1. Основы UI/UX-дизайна и работа с Figma		
	ОК 01, ОК 04 ПК 1.2, ПК 3.2	Тема 3.2. Итоговая работа		
<b>Уметь:</b> - создавать веб-страницы с использованием языка гипертекстовой разметки HTML; - оформлять веб-страницы с использованием каскадных таблиц стилей CSS; - разрабатывать адаптивные и кросс-браузерные интерфейсы;				

<ul style="list-style-type: none"> <li>- использовать инструменты разработчика для отладки кода;</li> <li>- работать с системами контроля версий (Git);</li> <li>- применять основы клиентского программирования на JavaScript;</li> <li>- использовать графические редакторы для создания макетов веб-страниц;</li> <li>- оптимизировать веб-страницы для поисковых систем (основы SEO).</li> </ul>				
--	--	--	--	--

## 2. Комплект оценочных средств

### 1. Задание для текущего контроля успеваемости

#### Раздел 1. Основы веб-технологий

#### Тема 1.1. Введение в веб-разработку

#### Практическое занятие №1:

«Настройка среды разработки. Анализ структуры HTTP-запросов с помощью инструментов разработчика»

#### Задание:

- 1) Установите и настройте редактор кода VS Code, установите необходимые расширения (Live Server, Prettier, HTML CSS Support).
- 2) Создайте простую HTML-страницу с любым содержимым.
- 3) Откройте страницу в браузере и с помощью инструментов разработчика (F12):

– Найдите вкладку Network и проанализируйте все запросы, которые отправляет страница.

– Изучите заголовки запроса и ответа для каждого ресурса.

– Определите код ответа сервера для каждого запроса.

- 4) Сделайте скриншоты и составьте отчет о структуре HTTP-запросов.

#### Практическое занятие №2:

«Исследование работы протоколов HTTP/HTTPS»

#### Задание:

- 1) Посетите сайты, работающие по HTTP и HTTPS.
- 2) С помощью инструментов разработчика сравните:
  - Наличие значка безопасности в адресной строке.
  - Заголовки ответа сервера (найдите заголовки, связанные с безопасностью).
  - Время загрузки страницы.
- 3) Объясните, почему использование HTTPS предпочтительнее.

Самостоятельная работа:

Подготовьте презентацию (5–7 слайдов) на тему «Эволюция веб-технологий и современные тренды». В презентации отразите:

- Основные этапы развития веба (Web 1.0, Web 2.0, Web 3.0).
- Современные тренды: PWA, SPA, JAMstack.
- Будущее веб-разработки.

**Тестовые задания:**

1. Что означает аббревиатура HTTP?

- a) HyperText Transfer Protocol
- b) High Technology Transfer Protocol
- c) HyperText Transmission Program
- d) Home Tool Transfer Protocol

2. Какой порт по умолчанию используется для HTTPS?

- a) 80
- b) 443
- c) 21
- d) 22

3. Что такое DNS?

- a) Система доменных имен
- b) Протокол передачи файлов
- c) Язык разметки
- d) Система управления базами данных

4. Какой код ответа HTTP означает "Успешно"?

- a) 200
- b) 404
- c) 500
- d) 301

5. Что такое хостинг?

- a) Услуга по предоставлению вычислительных мощностей для размещения сайта
- b) Имя сайта
- c) Протокол передачи данных
- d) Язык программирования

Тема 1.2. Системы контроля версий

Практическое занятие №1:

«Создание локального репозитория. Публикация проекта на GitHub»

Задание:

- 1) Установите Git на свой компьютер.
- 2) Настройте Git (укажите имя пользователя и email).
- 3) Создайте локальный репозиторий для папки с проектом.
- 4) Выполните инициализацию репозитория (git init).
- 5) Добавьте файлы в отслеживание (git add .) и создайте первый коммит (git commit -m "Initial commit").
- 6) Зарегистрируйтесь на GitHub и создайте удаленный репозиторий.

7) Свяжите локальный репозиторий с удаленным и отправьте изменения (git push).

Практическое занятие №2:

«Работа с ветками. Моделирование конфликта и его разрешение»

Задание:

- 1) Создайте новую ветку feature (git branch feature, git checkout feature).
- 2) Внесите изменения в файл проекта в ветке feature и сделайте коммит.
- 3) Переключитесь на ветку main и внесите изменения в ту же строку того же файла.
- 4) Выполните слияние ветки feature в main (git merge feature).
- 5) Разрешите возникший конфликт вручную, отредактировав файл.
- 6) Сделайте коммит с разрешением конфликта.

Самостоятельная работа:

Изучите дополнительные возможности Git: git stash, git rebase, git cherry-pick.

Подготовьте краткую шпаргалку по этим командам с примерами использования.

Тестовые задания:

1. Какая команда используется для создания коммита?
  - a) git commit
  - b) git push
  - c) git add
  - d) git init
2. Что делает команда git pull?
  - a) Загружает изменения из удаленного репозитория и сливает с локальным
  - b) Отправляет изменения в удаленный репозиторий
  - c) Создает новую ветку
  - d) Удаляет репозиторий
3. Как создать новую ветку с именем "develop"?
  - a) git branch develop
  - b) git checkout develop
  - c) git new develop
  - d) git create develop
4. Что такое конфликт в Git?
  - a) Ситуация, когда два разработчика изменили одну и ту же строку в файле
  - b) Ошибка подключения к интернету
  - c) Проблема с правами доступа
  - d) Отсутствие коммитов
5. Какой сервис является платформой для хостинга Git-репозитория?
  - a) GitHub
  - b) GitLab
  - c) Bitbucket
  - d) Все вышеперечисленные

Раздел 2. Клиентская веб-разработка

Тема 2.1. Язык гипертекстовой разметки HTML

Практическое занятие №1:

## «Создание структуры веб-страницы с использованием семантических тегов»

Задание:

- 1) Создайте HTML-страницу с семантической структурой:

html

```
<header>
  <nav>Меню навигации</nav>
</header>
<main>
  <article>
    <section>Раздел 1</section>
    <section>Раздел 2</section>
  </article>
  <aside>Боковая панель</aside>
</main>
<footer>Подвал сайта</footer>
```

- 2) Заполните страницу контентом (заголовки, абзацы, списки).

Практическое занятие №2:

«Создание многостраничного сайта с навигацией»

Задание:

- 1) Создайте три HTML-страницы: index.html, about.html, contacts.html.
- 2) На каждой странице создайте меню навигации со ссылками на все страницы.
- 3) Заполните страницы соответствующим контентом.
- 4) Добавьте на страницы изображения.

Практическое занятие №3:

«Разработка сложной формы регистрации/обратной связи»

Задание:

- 1) Создайте форму со следующими полями:
  - Имя (текстовое поле)
  - Email (поле для email)
  - Пароль (поле для пароля)
  - Подтверждение пароля
  - Пол (радиокнопки)
  - Страна (выпадающий список)
  - Согласие на обработку данных (чекбокс)
  - Кнопка отправки
- 2) Используйте семантическую разметку формы (fieldset, legend, label).

Практическое занятие №4:

«Встраивание мультимедиа-контента на веб-страницу»

Задание:

- 1) Добавьте на страницу аудио- и видеоэлементы с использованием тегов <audio> и <video>.
- 2) Вставьте карту Google Maps с помощью <iframe>.

3) Добавьте YouTube-видео через <iframe>.

Тестовые задания:

1. Какой тег используется для создания абзаца?

- a) <p>
- b) <h1>
- c) <div>
- d) <span>

2. Какой атрибут используется для указания адреса ссылки?

- a) src
- b) href
- c) link
- d) url

3. Какой тег является семантическим?

- a) <div>
- b) <span>
- c) <article>
- d) <b>

4. Какой тег используется для вставки изображения?

- a) <img>
- b) <image>
- c) <pic>
- d) <src>

5. Какой тег используется для создания нумерованного списка?

- a) <ul>
- b) <ol>
- c) <li>
- d) <list>

## Тема 2.2. Каскадные таблицы стилей CSS

### Практическое занятие №1:

«Оформление веб-страницы с использованием CSS: работа с типографикой и цветом»

Задание:

1) Подключите CSS к HTML-странице.

2) Задайте для страницы:

- Основной цвет фона и текста.
- Шрифты для заголовков и основного текста (используйте Google Fonts).
- Размеры и цвета заголовков.
- Межстрочный интервал, отступы для абзацев.

Практическое занятие №2:

«Верстка макета с использованием Flexbox»

Задание:

1) Создайте HTML-структуру для карточек товаров (заголовок, изображение, цена, кнопка).

2) С помощью Flexbox разместите 4 карточки в ряд.



- 3) Настройте отступы между карточками.
- 4) Сделайте так, чтобы при уменьшении экрана карточки переносились на следующую строку.

Практическое занятие №3:

«Верстка макета с использованием CSS Grid»

Задание:

- 1) Создайте макет страницы с помощью Grid:
  - Шапка (занимает всю ширину)
  - Основная область (3 колонки: сайдбар, основной контент, правый сайдбар)
  - Подвал (занимает всю ширину)
- 2) Заполните каждую область контентом.

Практическое занятие №4:

«Создание анимации и интерактивных элементов на CSS»

Задание:

- 1) Создайте кнопку с эффектами при наведении (изменение цвета, тени, масштаба).
- 2) Создайте простую CSS-анимацию (например, вращающийся элемент или плавно появляющийся блок).
- 3) Добавьте плавные переходы для всех интерактивных элементов.

Практическое занятие №5:

«Создание адаптивного меню и адаптация страницы под мобильные устройства»

Задание:

- 1) Создайте горизонтальное меню для десктопа.
- 2) Используя медиазапросы, превратите меню в вертикальное для мобильных устройств.
- 3) Адаптируйте карточки товаров: на десктопе 4 в ряд, на планшете 2 в ряд, на телефоне 1 в ряд.

Практическое занятие №6:

«Верстка сложного макета с использованием комбинации Flexbox и Grid»

Задание:

- 1) Создайте макет интернет-магазина, где:
  - Общая сетка страницы построена на Grid.
  - Карточки товаров в каталоге выровнены с помощью Flexbox.
  - Шапка и подвал используют Flexbox для расположения элементов.

Тестовые задания:

1. Какой селектор выбирает элементы по идентификатору?
  - a) .class
  - b) #id
  - c) element
  - d) \* (звездочка)
2. Какое свойство задает цвет текста?
  - a) background-color
  - b) color

c) font-color

d) text-color

3. Что такое Flexbox?

a) Одномерная система компоновки элементов

b) Двухмерная система компоновки

c) Язык программирования

d) Фреймворк для верстки

4. Какое свойство используется для создания отступа внутри элемента?

a) margin

b) padding

c) border

d) outline

5. Как сделать элемент блочным?

a) display: block

b) display: inline

c) display: none

d) visibility: hidden

6. Какое правило используется для создания адаптивного дизайна?

a) @media

b) @import

c) @font-face

d) @keyframes

7. Что делает свойство justify-content: center во Flexbox?

a) Выравнивает элементы по центру главной оси

b) Выравнивает элементы по центру поперечной оси

c) Задаёт отступы между элементами

d) Делает элементы невидимыми

8. Какой единицей измерения задается размер относительно размера экрана?

a) px

b) em

c) vw

d) pt

Тема 2.3. Основы клиентского программирования JavaScript

Практическое занятие №1:

«Основы программирования на JavaScript: работа с переменными, функциями»

Задание:

1) Напишите скрипт, который:

– Объявляет переменные разных типов (число, строка, булево значение, массив).

– Выводит значения в консоль.

– Создает функцию, которая принимает два числа и возвращает их сумму.

– Создает функцию, которая проверяет, является ли число четным.

Практическое занятие №2:

«Работа с массивами и объектами»

Задание:

- 1) Создайте массив из 5 товаров (каждый товар – объект с полями: название, цена, количество).
- 2) Напишите функцию, которая выводит список товаров на страницу.
- 3) Напишите функцию, которая подсчитывает общую стоимость всех товаров.
- 4) Напишите функцию, которая ищет товар по названию.

Практическое занятие №3:

«Обработка событий: динамическое изменение содержимого страницы»

Задание:

- 1) Создайте кнопку "Сменить тему", при клике на которую меняется цвет фона страницы.
- 2) Создайте кнопку "Показать/скрыть текст", которая показывает или скрывает блок с текстом.
- 3) Создайте список задач, куда можно добавлять новые элементы через форму.

Практическое занятие №4:

«Валидация формы с помощью JavaScript»

Задание:

- 1) Создайте форму регистрации (поля: имя, email, пароль, подтверждение пароля).
- 2) Напишите скрипт, который при отправке формы проверяет:
  - Имя не пустое.
  - Email имеет корректный формат.
  - Пароль содержит не менее 6 символов.
  - Пароль и подтверждение пароля совпадают.
- 3) При наличии ошибок выводите сообщения под соответствующими полями.

Практическое занятие №5:

«Создание динамического списка задач (ToDo-приложение)»

Задание:

- 1) Создайте интерфейс: поле ввода задачи, кнопка "Добавить", список задач.
- 2) При добавлении задача появляется в списке.
- 3) У каждой задачи должна быть кнопка "Выполнено" (зачеркивает задачу) и "Удалить".
- 4) Сохраняйте задачи в localStorage, чтобы они не исчезали после перезагрузки страницы.

Практическое занятие №6:

«Загрузка и отображение данных с использованием Fetch API»

Задание:

- 1) Используйте публичное API (например, JSONPlaceholder) для получения списка постов.
- 2) Загрузите данные с помощью fetch().
- 3) Отобразите полученные посты на странице в виде карточек.
- 4) Добавьте индикатор загрузки (спиннер), пока данные загружаются.

Тестовые задания:

1. Как объявить переменную в JavaScript?
  - a) var name
  - b) let name
  - c) const name
  - d) Все вышеперечисленное
2. Что выведет console.log(typeof 42)?
  - a) "number"
  - b) "string"
  - c) "boolean"
  - d) "object"
3. Какой метод добавляет элемент в конец массива?
  - a) push()
  - b) pop()
  - c) shift()
  - d) unshift()
4. Что такое DOM?
  - a) Объектная модель документа
  - b) Язык разметки
  - c) База данных
  - d) Протокол передачи данных
5. Как выбрать элемент по его ID в JavaScript?
  - a) document.getElementById('id')
  - b) document.querySelector('#id')
  - c) Оба варианта работают
  - d) Ни один из вариантов
6. Что делает метод addEventListener?
  - a) Добавляет обработчик события
  - b) Удаляет обработчик события
  - c) Создает новое событие
  - d) Вызывает событие
7. Какая функция используется для отправки HTTP-запросов?
  - a) fetch()
  - b) request()
  - c) http()
  - d) ajax()
8. Что такое Promise?
  - a) Объект, представляющий результат асинхронной операции
  - b) Тип данных
  - c) Функция
  - d) Массив

Раздел 3. Инструменты дизайна и итоговый проект

Тема 3.1. Основы UI/UX-дизайна и работа с Figma

Практическое занятие №1:

«Создание вайрфрейма главной страницы сайта в Figma»

Задание:

- 1) Зарегистрируйтесь в Figma.
- 2) Создайте новый файл и фрейм для десктопа (1440×1024).
- 3) Разработайте вайрфрейм главной страницы интернет-магазина:
  - Шапка (логотип, меню, корзина)
  - Баннер
  - Каталог товаров (сетка из карточек)
  - Подвал
- 4) Используйте только прямоугольники и текст, без цветов и детальной

графики.

Практическое занятие №2:

«Разработка интерактивного прототипа в Figma»

Задание:

- 1) На основе вайрфрейма создайте интерактивный прототип:
  - При клике на ссылку меню происходит переход на соответствующий фрейм.
  - При клике на кнопку "Купить" появляется всплывающее окно (pop-up).
- 2) Настройте анимацию переходов.

Самостоятельная работа:

Создайте макет для итогового проекта (главная страница + 2 внутренних).

Тема 3.2. Итоговая работа

Практическое занятие №1:

«Разработка многостраничного веб-сайта на основе собственного макета»

Задание:

- 1) Разработайте концепцию сайта (тематика: интернет-магазин, блог, портфолио, корпоративный сайт).
- 2) Создайте макет сайта в Figma (главная страница + минимум 2 внутренних страницы).
- 3) Сверстайте сайт по макету, используя HTML и CSS.
- 4) Добавьте интерактивность с помощью JavaScript.
- 5) Обеспечьте адаптивность верстки.
- 6) Загрузите проект в репозиторий Git.

Практическое занятие №2:

«Защита итогового проекта»

Задание:

- 1) Подготовьте презентацию (5–7 слайдов) с описанием проекта:
  - Цель и задачи
  - Макеты в Figma
  - Особенности реализации
  - Демонстрация рабочего сайта
- 2) Презентуйте проект, ответьте на вопросы.

Самостоятельная работа (подготовка к экзамену):

- Разработка концепции и макета итогового проекта.
- Верстка и программирование итогового проекта.
- Подготовка презентации для защиты проекта.

- Повторение теоретического материала.

## 2. Вопросы и задание для промежуточной аттестации

1. Архитектура веб-приложений: клиент-серверное взаимодействие.
2. Протоколы HTTP/HTTPS: структура запроса и ответа, коды состояния.
3. Понятие домена, хостинга, DNS.
4. Системы контроля версий. Git: основные команды и принципы работы.
5. Ветвление в Git: создание веток, слияние, разрешение конфликтов.
6. Удаленные репозитории (GitHub, GitLab): работа с ними.
7. Структура HTML-документа. Семантическая верстка.
8. Основные теги HTML: заголовки, абзацы, списки, ссылки, изображения, таблицы.
9. Формы в HTML: элементы ввода, их типы и атрибуты.
10. Способы подключения CSS. Селекторы в CSS.
11. Блочная модель документа (box model). Позиционирование элементов.
12. Flexbox: свойства контейнера и элементов.
13. CSS Grid: построение двумерных сеток.
14. Адаптивная верстка: медиазапросы.
15. Основы JavaScript: переменные, типы данных, операторы.
16. Управляющие конструкции: условные операторы, циклы.
17. Функции в JavaScript: объявление, параметры, возврат значения.
18. Массивы и объекты в JavaScript.
19. События и обработчики событий. Взаимодействие с DOM.
20. Валидация форм на стороне клиента.
21. Асинхронность в JavaScript: Promise, async/await, Fetch API.
22. Понятие UI/UX. Принципы проектирования пользовательских интерфейсов.
23. Создание макетов в Figma: вайрфреймы, прототипы, компоненты.
24. Подготовка графики для веб: оптимизация изображений, выбор формата.
25. Основы оптимизации веб-страниц и SEO.