



Основная профессиональная образовательная программа  
02.03.02 Фундаментальная информатика и информационные технологии  
(Программирование и информационные технологии)

Министерство науки и высшего образования Российской Федерации

**ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Кафедра информационных технологий и прикладной математики

ОДОБРЕНО:

Руководитель ОП

\_\_\_\_\_ С.В. Данилова  
(подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**Рабочая программа дисциплины**

**Web-программирование**

Уровень высшего образования:	бакалавриат
Квалификация выпускника:	бакалавр
Направление подготовки:	02.03.02      Фундаментальная      информатика      и информационные технологии
Направленность (профиль) образовательной программы:	Программирование и информационные технологии



## 1. Цели освоения дисциплины

Цель освоения дисциплины «Web-программирование» — формирование у студентов компетенций, необходимых для создания клиентской части (frontend) современных, адаптивных и интерактивных веб-приложений, начиная с фундаментальных основ верстки и заканчивая использованием прогрессивных фреймворков.

Задачи дисциплины:

1. Сформировать устойчивые навыки создания семантической и валидной HTML-разметки как основы любого веб-интерфейса.
2. Овладеть современными технологиями CSS, включая Flexbox, Grid Layout и медиазапросы, для построения сложных, адаптивных и кросс-браузерных пользовательских интерфейсов.
3. Изучить базовые принципы и синтаксис JavaScript для добавления интерактивности на веб-страницы, манипуляции с DOM-элементами и обработки событий.
4. Освоить концепции компонентного подхода и реактивности на примере фреймворка Vue.js для построения структурированных и легко масштабируемых веб-приложений.
5. Приобрести практический опыт сквозной разработки от статического макета до работающего прототипа веб-приложения, включая этапы верстки, добавления логики и интеграции с внешними API.
6. Развить навыки работы с современным инструментарием разработчика (Vite, Git, DevTools) и понимание принципов кросс-браузерного и адаптивного тестирования.

## 2. Место дисциплины в структуре ОП

Дисциплина «Web-программирование» относится к части образовательной программы, формируемой участниками образовательных отношений.

Успешное освоение данной дисциплины будет способствовать готовности студентов для освоения следующих дисциплин: Электронный документооборот на предприятии, Технологическое и социальное предпринимательство, Разработка образовательных ресурсов, Коммерциализация интеллектуальной собственности, а также прохождения преддипломной практики и подготовки ВКР. Также полученные знания, умения и навыки могут быть использованы бакалавром фундаментальной информатики в своей профессиональной деятельности и при обучении в магистратуре.

Студент, приступающий к изучению дисциплины, должен обладать следующими знаниями, умениями, навыками, полученными ранее в ходе изучения дисциплин: Операционные системы, Языки программирования, Базы данных, Технологии разработки программных приложений, Сети и системы передачи информации.

Для освоения данной дисциплины обучающийся должен:

Знать:

- операционные системы, базовое и системное программное обеспечение;
- методы и принципы управления требованиями пользователей;
- жизненный цикл ИС;
- основы информационной безопасности;
- выявление степени и необходимости переработки ИС;
- этапы разработки ПО и ИС;



- технические платформы.

Уметь:

- пользоваться основным инструментарием проектирования ИС;
- производить экономическую оценку принимаемых решений;
- определять их объем и трудозатраты по интеграции компонентов ИС.

Иметь:

- иметь навыки проектирования ИС;
- иметь опыт работы с базовым программным обеспечением и средствами вычислительной техники.

### **3. Планируемые результаты обучения по дисциплине**

#### **3.1. Компетенции, формированию которых способствует дисциплина**

При освоении дисциплины формируются следующие компетенции в соответствии с ФГОС ВО по данному направлению подготовки:

- профессиональные (ПК):

ПК-3: Способен проводить работы по проектированию программного обеспечения

#### **3.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения формируемых компетенций**

В результате освоения дисциплины обучающийся должен:

Знать:

- принципы и паттерны проектирования пользовательских интерфейсов (UI) и взаимодействия (UX) (ПК-3);
- архитектурные подходы к построению клиентской части приложений: от монолитной верстки до компонентной архитектуры (Vue.js) (ПК-3);
- критерии выбора технологий (HTML5, CSS3, JS, фреймворки) для решения различных классов задач frontend-разработки (ПК-3);
- модели и стандарты (семантическая верстка, RESTful API) для обеспечения взаимодействия между клиентом и сервером (ПК-3).

Уметь:

- анализировать требования к программному обеспечению и проектировать структуру HTML-документа на основе семантических принципов (ПК-3);
- разрабатывать макеты и проектировать адаптивную систему компонентов интерфейса с использованием современных CSS-технологий (Flexbox, Grid) (ПК-3);
- моделировать данные на клиентской стороне и проектировать их взаимодействие с пользовательским интерфейсом (ПК-3);
- проектировать компонентную структуру frontend-приложения на основе фреймворка (Vue.js), определяя иерархию компонентов, потоки данных (props/events) и состояние приложения (ПК-3).

Иметь:

- практический опыт декомпозиции сложного пользовательского интерфейса на составляющие компоненты и элементы (ПК-3);



Основная профессиональная образовательная программа  
02.03.02 Фундаментальная информатика и информационные технологии  
(Программирование и информационные технологии)

- навыки проектирования и прототипирования интерфейсов с помощью инструментов (Figma, диаграммы) и последующей реализации макетов в коде (ПК-3);
- практический опыт формализации предметной области (данные, с которыми работает интерфейс) в виде моделей, пригодных для реализации на JavaScript/Vue.js (ПК-3);
- навыки чтения, анализа и использования технической документации (API-документы, спецификации фреймворков) для проектных решений (ПК-3).

#### 4. Объем и содержание дисциплины

Объем дисциплины составляет 3 зачетных единиц (108 академических часов).

##### 4.1. Содержание дисциплины по разделам (темам), соотнесенное с видами и трудоемкостью занятий лекционно-семинарского типа

Объем иной контактной работы и самостоятельной работы обучающегося по дисциплине указан в учебном плане образовательной программы.

№ п/п	Разделы (темы) дисциплины	Семестр	Виды занятий, их объем (в ак. часах, по очной форме обучения)		Формы текущего контроля успеваемости (по очной форме обучения)
			Занятия лекцион- ного типа	Занятия семинар- ского типа	Формы промежуточной аттестации
Раздел 1. Основы веб-верстки (HTML5 и CSS3)					
1.	Введение в HTML. Семантическая разметка.	5	2	8	Лабораторная работа
2.	Основы CSS. Блочная модель и позиционирование.	5	2	8	Лабораторная работа
3.	Современные макеты: Flexbox.	5	2	8	Лабораторная работа
4.	Современные макеты: CSS Grid Layout.	5	2	8	Лабораторная работа
Раздел 2. Адаптивность и инструменты					
5.	Адаптивный и отзывчивый дизайн.	5	2	4	Лабораторная работа
6.	Препроцессор SASS/SCSS.	5	2	2	Лабораторная работа
Раздел 3. Интерактивность и фреймворки					
7.	Основы JavaScript для фронтенд-разработки.	5	2	4	Лабораторная работа
8.	Фреймворк Vue.js (Composition API).	5	4	6	Лабораторная работа
Итого за семестр:			18	48	Зачет
Итого по дисциплине:			18	48	Зачет

##### 4.2. Развернутое описание содержания дисциплины по разделам (темам)

###### *Раздел 1. Основы веб-верстки (html5 и css3)*

Цель раздела: сформировать у студентов фундаментальное понимание структуры веб-страницы и принципов её визуального представления.

1.1. Введение в HTML. Семантическая разметка. Основы клиент-серверного взаимодействия. Роль HTML, CSS, JavaScript. Структура HTML-документа. Обзор основных тегов (<h1>-<h6>, <p>, <ul>, <ol>, <li>, <a>, <img>). Понятие семантической верстки и её



значение для доступности и SEO. Семантические теги (<header>, <nav>, <main>, <section>, <article>, <footer>, <aside>).

1.2. Основы CSS. Блочная модель и позиционирование. Способы подключения CSS к HTML. Селекторы (тег, класс, id, атрибут). Каскадность и специфичность. Свойства для работы с текстом (font-\*, text-\*, color). Блочная модель: width, height, margin, padding, border, box-sizing. Модели позиционирования: static, relative, absolute, fixed, sticky.

1.3. Современные макеты: Flexbox. Проблемы верстки до Flexbox. Концепция Flex-контейнера и Flex-элементов. Основные свойства контейнера: display: flex, flex-direction, justify-content, align-items, flex-wrap, gap. Свойства элементов: align-self, order, flex-grow, flex-shrink.

1.4. Современные макеты: CSS Grid Layout. Концепция двумерной сетки (Grid). Определение сетки: grid-template-columns, grid-template-rows, grid-template-areas. Размещение элементов: grid-column, grid-row, grid-area. Выравнивание: justify-items, align-items, gap. Сочетание Grid (для общего макета страницы) и Flexbox (для внутреннего выравнивания компонентов).

### **Раздел 2. Адаптивность и инструменты**

Цель раздела: научить создавать интерфейсы, корректно отображающиеся на любых устройствах, и познакомить с инструментами для эффективной работы.

2.1. Адаптивный и отзывчивый дизайн. Понятия Adaptive и Responsive Web Design. Мета-тег viewport. Медиазапросы (@media): синтаксис, логические операторы, типы медиаустройств. Ключевые контрольные точки (breakpoints). Подход Mobile First. Респонсивные изображения (max-width: 100%, <picture>, srcset). Относительные единицы измерения: %, vw, vh, rem, em.

2.2. Препроцессор SASS/SCSS. Преимущества использования препроцессоров. Переменные для хранения цветов, шрифтов, размеров. Вложенность селекторов. Миксины (Mixins) для переиспользуемых стилей. Фрагментация кода и директива @import/@use.

### **Раздел 3. Интерактивность и фреймворки**

Цель раздела: научить добавлять динамическое поведение на страницу с помощью JavaScript и познакомить с принципами разработки на современном фреймворке Vue.js.

3.1. Основы JavaScript для фронтенд-разработки. Базовый синтаксис: переменные (const, let), типы данных, операторы, условия, циклы. Функции (Function Declaration, Arrow Functions). Объектная модель документа (DOM): дерево узлов. Поиск элементов (querySelector, querySelectorAll). Манипуляция содержимым, атрибутами и стилями. Обработка событий: addEventListener, объект события, основные типы событий (click, submit, input).

3.2. Фреймворк Vue.js (Composition API). Введение в концепцию реактивности. Экземпляр приложения Vue. Основы Composition API: ref(), reactive(). Директивы: v-bind, v-model, v-if/v-else, v-for. Обработка событий (v-on). Компонентный подход. Создание и регистрация компонентов. Передача данных: Props (вниз) и Emits (вверх). Жизненный цикл компонента. Слоты (<slot>).

## **5. Образовательные технологии**

Организация учебного процесса по дисциплине «Web-программирование» осуществляется в формате смешанного обучения (blended learning) и сочетает традиционные формы (лекции, лабораторные занятия) с современными интерактивными и цифровыми технологиями, направленными на формирование практико-ориентированных компетенций.

В учебном процессе используются следующие образовательные технологии:

– Проектно-ориентированное обучение: сквозной учебный проект является ядром курса. С первого занятия студенты начинают работу над собственным веб-приложением (например, интернет-магазин, личный кабинет, интерактивное портфолио), поэтапно применяя



Основная профессиональная образовательная программа  
02.03.02 Фундаментальная информатика и информационные технологии  
(Программирование и информационные технологии)

---

полученные знания: от верстки макета до внедрения интерактивности на Vue.js. Это обеспечивает целостное понимание процесса разработки.

- Технология поэтапного погружения: курс построен по принципу «от простого к сложному»: HTML → CSS → JavaScript → Vue.js. Каждая последующая тема логически вытекает из предыдущей и демонстрирует ее необходимость (например, ограничения нативного JS показывают потребность во фреймворке).

- Интерактивные лекции и воркшопы: лекционный материал подается не в чисто теоретическом, а в демонстрационном формате («live coding»). Преподаватель параллельно с объяснением пишет код, который студенты сразу могут повторять и экспериментировать с ним. Это превращает лекцию в интерактивный воркшоп.

- Проблемно-ориентированное обучение: Лабораторные работы строятся вокруг решения конкретных практических проблем:

- «Как сверстать адаптивную карточку товара?»
- «Как реализовать добавление товара в корзину без перезагрузки страницы?»
- «Как организовать код сложного приложения, чтобы его было легко поддерживать?»

- Студенты учатся самостоятельно искать решения, используя официальную документацию (MDN, Vue.js Guide), форумы (Stack Overflow) и профессиональные сообщества.

- Парное программирование: на некоторых лабораторных занятиях студенты объединяются в пары («водитель» + «штурман») для совместного решения задач. Это развивает навыки командной работы, код-ревью и объяснения своих решений.

- Технологии тестового контроля: для оперативной проверки усвоения теоретического материала (синтаксис, основные понятия) используются онлайн-тесты (Google Forms, OnlineTestPad, Kahoot) в начале или в конце занятия.

В перечень информационных технологий, используемых при осуществлении образовательного процесса, входят:

- Технологии смешанного обучения: ЭИОС «Мой университет» (или аналог) для размещения лекционных материалов, лабораторных заданий, видео-тutorиалов, проведения тестов и организации обсуждений.

- Средства разработки: Современная редакция кода Visual Studio Code с обязательным использованием плагинов для продуктивной работы (Live Server, Prettier, ESLint, Vetur/Volar).

- Система управления версиями: Git и GitHub/GitLab для ведения истории кода учебного проекта и обучения работе в распределенной команде.

- Мультимедиа технологии.

- Онлайн-инструменты для проверки кода: Сервисы для проверки валидности HTML/CSS-кода, проверки адаптивности (Chrome DevTools Device Mode), скорости загрузки.

- Low-Code/No-Code платформы: Использование Figma для работы с готовыми макетами, которые студенты будут верстать, что имитирует реальные условия работы с дизайнером.



## **6. Учебно-методическое обеспечение самостоятельной работы обучающихся**

Для дисциплины предусмотрен один вид самостоятельной работы. Проработка лекционного материала в виде самостоятельной работы над практическими заданиями. Выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию. А также дорабатывается самостоятельно.

К зачету допускаются студенты, которые систематически, в течение всего семестра работали на занятиях и показали уверенные знания по вопросам, выносившимся на практические занятия.

Непосредственная подготовка к зачету осуществляется по вопросам, представленным в приложении к РПД на основе МУ (приложение №1). Материалы находятся в электронной образовательной среде «Мой университет» (<https://uni.ivanovo.ac.ru>), а также на сетевом диске в соответствующей папке дисциплины.

## **7. Характеристика оценочных средств для текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине**

Оценочные средства включают средства входного, промежуточного и выходного контроля, а также описания форм и процедур, предназначенных для определения качества освоения обучающимися учебного материала, учебной дисциплины, профессионального модуля, направленные на измерение степени сформированности компетенции, как в целом, так и отдельных ее компонентов.

Оценка компетентности осуществляется на основе защиты практических работ и сдачи устного зачета. В процессе выполнения практических работ и устного зачета выявляется сформированность компетенций.

Зачет на основе текущего контроля может быть выставлен на основе выполнения и защиты всех практических работ.

Для сдачи зачета в устной форме необходимо сдать зачет по вопросам – преподавателем задается два из вопроса из списка, отводится время на подготовку не более 20 мин. Зачет сдается в виде устного ответа на вопрос, в процессе ответа преподаватель задает по мере необходимости дополнительные вопросы, позволяющие ему оценивать уровень подготовки студента.

Оценка «зачтено» ставится при выполнении следующего условия:

Ответы на поставленные вопросы излагаются логично, последовательно и не требуют дополнительных пояснений. Делаются обоснованные выводы. Демонстрируются глубокие знания базовых нормативно-правовых актов. Соблюдаются нормы литературной речи. Ответы на поставленные вопросы излагаются систематизировано и последовательно. Материал излагается уверенно. Демонстрируется умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер. Соблюдаются нормы литературной речи. Хороший уровень знаний в области ИС.

Оценка «незачтено» ставится при следующих условиях:

Допускаются нарушения в последовательности изложения. Демонстрируются поверхностные знания вопроса. Имеются затруднения с выводами. Отмечается слабое владение теоретическими основами. Материал излагается непоследовательно, сбивчиво, не представляет определенной системы знаний по дисциплине. Имеются заметные нарушения норм литературной речи. «Незачтено» выставляется студенту, обнаружившему пробелы в знаниях основного программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий.

Оценочные средства дисциплины представлены в приложении 2.



## 8. Учебно-методическое и информационное обеспечение дисциплины

### Основная литература:

1. Минязев Р. Ш. Скриптовые языки web-программирования (JavaScript, PHP, html/CSS): учебно-методическое пособие / Р. Ш. Минязев. — Казань: КНИТУ-КАИ, 2022. — 60 с. — ISBN 978-5-7579-2632-2. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/399557> — Режим доступа: для авториз. пользователей.
2. Петракова Н. В. Основы HTML: учебно-методическое пособие / Н. В. Петракова. — Брянск: Брянский ГАУ, 2022 — Часть 1 — 2022. — 50 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/304958> — Режим доступа: для авториз. пользователей.
3. Никулова Г. А. Web-технологии: Введение в программирование на JavaScript: защита контента средствами JS и CSS: учебно-методическое пособие / Г. А. Никулова, А. С. Терлецкий. — Липецк: Липецкий ГПУ, 2023. — 76 с. — ISBN 978-5-907792-00-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/403757> — Режим доступа: для авториз. пользователей.
4. Никулова Г. А. Web-дизайн. Приемы адаптивного Web-дизайна: технологии Flexbox и CSS Grid: учебное пособие / Г. А. Никулова, А. С. Терлецкий. — Липецк: Липецкий ГПУ, 2021. — 69 с. — ISBN 978-5-907461-41-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/228698> — Режим доступа: для авториз. пользователей.

### Дополнительная литература:

1. WEB-девелопмент и WEB-дизайн в электронном бизнесе: учебное пособие. — Санкт-Петербург: СПбГУТ им. М.А. Бонч-Бруевича, 2017 — Часть 1 — 2017. — 90 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/180259> — Режим доступа: для авториз. пользователей.
2. WEB-девелопмент и WEB-дизайн в электронном бизнесе: учебное пособие. — Санкт-Петербург: СПбГУТ им. М.А. Бонч-Бруевича, 2017 — Часть 2 — 2017. — 95 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/180260> — Режим доступа: для авториз. пользователей.
3. Кожевникова П. В. PHP и MySQL: учебное пособие / П. В. Кожевникова. — Ухта: УГТУ, 2020. — 51 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/209591> — Режим доступа: для авториз. пользователей.
4. Попов С. Е. JavaScript. Основы программирования: учебно-методическое пособие / С. Е. Попов. — Елец: ЕГУ им. И.А. Бунина, 2020. — 121 с. — ISBN 978-5-00151-175-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/331940> — Режим доступа: для авториз. пользователей.
5. Основы Web-дизайна: учебно-методическое пособие / составитель Н. А. Саблина. — Липецк: Липецкий ГПУ, 2018. — 50 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/115017> — Режим доступа: для авториз. пользователей.

### Ресурсы информационно-телекоммуникационной сети «Интернет»:

Система электронной поддержки образовательного процесса «Мой университет»  
<https://uni.iivanovo.ac.ru>

### Профессиональные базы данных и информационно-справочные системы:

ЭБС «Университетская библиотека онлайн» [www.biblioclub.ru](http://www.biblioclub.ru)





Основная профессиональная образовательная программа  
02.03.02 Фундаментальная информатика и информационные технологии  
(Программирование и информационные технологии)

---

Электронная библиотека ИвГУ <http://lib.ivanovo.ac.ru>

Электронный каталог НБ ИвГУ <http://lib.ivanovo.ac.ru/index.php/ek>

База знаний IT-компании GeekBrains <https://gb.ru/posts>

Серия вебинаров по мобильной и кроссплатформенной разработке от компании Skillbox  
<https://live.skillbox.ru/code/?topics=mobilnye-prilozheniya&topics=krosplatformennaya-razrabotka>

Программное обеспечение: операционная система Microsoft Windows, пакет офисных программ Microsoft Office и(или) LibreOffice, интернет-браузер Microsoft Edge и(или) Yandex Browser и(или) Google Chrome, Visual Studio Code.

## **9. Материально-техническое обеспечение дисциплины**

Учебные аудитории:

- для проведения занятий лекционного типа с комплектом специализированной учебной мебели и техническими средствами обучения, служащими для предоставления учебной информации большой аудитории;

- для проведения занятий семинарского типа, консультаций, текущего контроля и промежуточной аттестации с комплектом специализированной учебной мебели и техническими средствами обучения.

Помещение для самостоятельной работы, оснащенное комплектом специализированной учебной мебели, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в ЭИОС.

Демонстрационное оборудование и учебно-наглядные пособия для занятий лекционного типа, обеспечивающие тематические иллюстрации: демонстрационные устройства; электронные пособия, печатные пособия.



Основная профессиональная образовательная программа  
02.03.02 Фундаментальная информатика и информационные технологии  
(Программирование и информационные технологии)

---

**Автор рабочей программы дисциплины:** ст. преподаватель каф. ИТиПМ Сидорова А.Д.

Программа рассмотрена и утверждена на заседании кафедры Информационных технологий и прикладной математики (ИТиПМ) «29» августа 2025 г., протокол № 1

Программа обновлена  
протокол заседания кафедры № \_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

Согласовано:

Руководитель ОП \_\_\_\_\_ Данилова С. В.

(подпись)

Программа обновлена  
протокол заседания кафедры № \_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

Согласовано:

Руководитель ОП \_\_\_\_\_ Данилова С. В.

(подпись)

Программа обновлена  
протокол заседания кафедры № \_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

Согласовано:

Руководитель ОП \_\_\_\_\_ Данилова С. В.

(подпись)