

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию
в области информатики и радиоэлектроники

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

_____ А.Г.Баханович

Регистрационный № _____

ВЕБ-ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

**Примерная учебная программа по учебной дисциплине
для специальности
6-05-0611-04 Электронная экономика**

СОГЛАСОВАНО

Председатель Учебно-методического
объединения по образованию в
области информатики и
радиоэлектроники

_____ В.А. Богуш

СОГЛАСОВАНО

Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь

_____ С.Н. Пищов

СОГЛАСОВАНО

Проректор по научно-методической
работе Государственного учреждения
образования «Республиканский
институт высшей школы»

_____ И.В. Титович

Эксперт-нормоконтролер

Минск 2025

СОСТАВИТЕЛЬ:

Н.А.Атрощенко, старший преподаватель кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»;

А.А.Ефремов, заведующий кафедрой экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат экономических наук, доцент

РЕЦЕНЗЕНТЫ:

Кафедра информационных технологий в экологии и медицине учреждения образования «Международный государственный экологический институт им.А.Д.Сахарова» Белорусского государственного университета (протокол №9а от 12.05.2025);

Н.В.Шалькевич, директор общества с ограниченной ответственностью «Трисофт»

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ПРИМЕРНОЙ:

Кафедрой экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 15 от 02.05.2025);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № ____ от _____);

Научно-методическим советом по прикладным информационным системам и технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 13 от 04.06.2025)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Примерная учебная программа по учебной дисциплине «Веб-технологии и программирование» разработана для студентов учреждений высшего образования, обучающихся по специальности 6-05-0611-04 «Электронная экономика» в соответствии с требованиями образовательного стандарта общего высшего образования и примерного учебного плана вышеуказанной специальности.

Учебная дисциплина «Веб-технологии и программирование» является базовой в цикле дисциплин, ориентированных на создание клиент-серверных приложений для решения широкого круга задач, связанных с разработкой веб-проектов для решения профессиональных задач. Изучение настоящей дисциплины обеспечивает подготовку специалиста, владеющего фундаментальными знаниями и практическими навыками в области разработки веб-приложений на основе современных технологий.

Воспитательное значение учебной дисциплины «Веб-технологии и программирование» заключается в формировании у обучающихся математической культуры и научного мировоззрения; развитии исследовательских умений, аналитических способностей, креативности, необходимых для решения научных и практических задач; развитии познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности; формировании способностей к саморазвитию, самосовершенствованию и самореализации.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

ЦЕЛЬ, ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины: приобретение знаний и навыков разработки, настройки и конфигурирования полнофункциональных веб-приложений с использованием языка JavaScript и платформы Node.js, развитие профессиональных компетенций в области современных информационных технологий, практических навыков современного веб-разработчика.

Задачи учебной дисциплины:

овладение навыками разработки, модернизации, внедрения и использования информационных систем, технологий программирования в профессиональной деятельности;

изучение принципов создания и программной реализации веб-приложений с использованием современных программных средств и технологий;

овладение знаниями о системах и назначении стандартов проектирования, навыками разработки и обоснования выбора проектных решений;

приобретение навыков построения полноценных клиент-серверных приложений на основе современных архитектурных решений и паттернов проектирования;

изучение принципов работы веб-приложений на клиентской стороне и на стороне веб-сервера;

освоение навыков применения специализированных методов для построения веб-ориентированных информационных систем;

овладение основными инструментами и технологиями для построения пользовательских интерфейсов;

приобретение знаний в области теоретических и практических аспектов веб-программирования;

овладение методами использования современных инструментальных средств веб-разработки;

изучение принципов взаимодействия клиента и сервера с методами и процедурами конструирования и оптимизации программного кода.

Базовыми учебными дисциплинами по курсу «Веб-технологии и программирование» являются такие учебные дисциплины как «Основы алгоритмизации и программирования», «Технологии программирования». В свою очередь учебная дисциплина «Веб-технологии и программирование» является основой для таких учебных дисциплин, как «Информационные системы корпоративного управления» и «Электронный бизнес».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения дисциплины «Веб-технологии и программирование» формируется следующая базовая профессиональная компетенция: использовать методы и приемы разработки программ на основе шаблонов и библиотек классов, методы обработки и способы реализации программ в распределенных системах, проводить отладку и тестирование веб-прикладных программ.

В результате изучения учебной дисциплины обучающийся должен:

знать:

технологии разметки HTML-документа, средства HTML5 для формирования веб-страниц;

основы современных видов верстки дизайна веб-страниц;

основные элементы каскадных таблиц стилей (CSS), методы формирования стилевого решения с использованием возможностей CSS3, виды селекторов;

механизмы создания интерактивных динамических веб-страниц с использованием DOM, JavaScript, DHTML и современных JavaScript библиотек;

- возможности сохранения данных на клиентской стороне;
- основные принципы построения полноценных клиент-серверных приложений с учетом организации его информационных, вычислительных, валидационных и других составляющих;
- способы интеграции клиент-серверных веб-приложений с различными СУБД;
- основные характеристики, механизмы и возможности приложений клиент-серверной архитектуры;
- базовые концепции асинхронных клиент-серверных взаимодействий и основные подходы к построению интерактивных пользовательских интерфейсов веб-приложений на основе Ajax;
- основные применяемые и перспективные направления развития в области модульной разработки современных веб-проектов;
- принципы использования архитектурных решений и паттернов проектирования веб-приложений;
- современные средства, методы и технологии, применяемые в области программирования сложных пользовательских интерфейсов с высокой производительностью;
- уметь:*
 - проектировать клиент-серверные веб-приложения на основе языка JavaScript и платформы Node.js с применением современных подходов и технологических программных средств;
 - разрабатывать интерактивные динамические пользовательские интерфейсы;
 - применять актуальные подходы и средства при разработке веб-дизайна;
 - определять и осуществлять основные этапы разработки веб-приложения;
 - разрабатывать и развертывать базовые элементы клиентских, серверных и веб-приложений в рамках рассматриваемых технологий;
 - моделировать и обосновывать принимаемые решения на основе современных паттернов проектирования и архитектурных стандартов, используемых в веб-технологиях;
 - использовать асинхронный подход к построению интерактивных пользовательских интерфейсов веб-приложений;
 - применять серверные компоненты, актуальные библиотеки и технологии при проектировании веб-систем;
 - использовать модульный и комплексный подход для построения архитектурных решений веб-проектов;
 - применять современные программные средства (пакетные менеджеры, сборщики, загрузчики модулей и другие актуальные инструменты) при проектировании веб-приложений;
 - осуществлять интеграцию веб-приложений с хранилищами данных как на клиентской, так и на серверной стороне;
 - использовать современные системы автоматизации, моделирования, проектирования, сборки, развертывания и оптимизации веб-приложений;

иметь навык:

создания динамических веб-страниц с использованием HTML и DHTML, встраиваемого (и выполняемого на стороне клиента) скриптового языка JavaScript, CSS (каскадных таблиц стилей) и DOM (объектной модели документа);

работы с современными методами, технологиями и инструментальными средствами проектирования и разработки веб-приложений;

разработки полноценных клиент-серверных приложений на основе языка JavaScript и платформы Node.js

использования клиент-серверных технологий при создании веб-приложений на основе модульных концепций и шаблонов проектирования;

построения интерактивных пользовательских интерфейсов веб-приложений на основе вспомогательных средств и актуальных библиотек JavaScript;

интеграции веб-приложений с хранилищами данных и осуществления хранения данных как на клиентской, так и на серверной стороне;

использования асинхронного подхода к построению веб-приложений;

построения веб-приложения с использованием современных технических и инструментальных средств разработки, сборки, тестирования, развертывания и обновления веб-приложений.

Примерная учебная программа рассчитана на 288 учебных часов, из них – 132 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекции – 56 часов, лабораторные занятия – 76 часов.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 1. Создание, верстка и стилевое оформление веб-страниц с размещением гипертекстовой информации на основе структурированных документов	48	16	32
Тема 1. Гипертекстовый язык HTML. Основные определения и характеристики веб-страницы	2	2	-
Тема 2. Формирование статического контента веб-страницы. Размещение веб-элементов на странице	6	2	4
Тема 3. Каскадные таблицы стилей CSS. Объявление стилей	6	2	4
Тема 4. Виды селекторов. Разработка дизайна веб-страниц на основе категорий свойств и типов селекторов CSS. Стандарт CSS3 и его возможности	6	2	4
Тема 5. Создание структуры гипертекстового документа на основе HTML-разметки и блочной верстки макетов веб-страниц	6	2	4
Тема 6. Современные подходы к системе компоновки элементов на веб-странице и стилизации размещения веб-элементов	6	2	4
Тема 7. Объектная модель документа DOM (Document Object Model) и объектная модель браузера BOM (Browser Object Model). Возможность доступа к веб-элементам и узлам на странице	4	-	4
Тема 8. Формирование интерактивных динамических веб-страниц с помощью динамического доступа к содержимому документа, его структуре и стилям с помощью DHTML. Обработка событий в JavaScript	6	2	4
Тема 9. Применение объектной модели документов для работы с формами ввода данных	6	2	4
Раздел 2. Введение в клиентские сетевые технологии. Основы языка JavaScript	28	12	16
Тема 10. Введение в мультипарадигменный язык программирования JavaScript	2	2	-
Тема 11. Основные конструкции языка JavaScript. Массивы. Управляющие структуры и организация циклов. Функции. Обработка событий в JavaScript	6	2	4

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Тема 12. Понятие объекта JavaScript. Стандартные встроенные объекты и их назначение	2	2	-
Тема 13. Организация работы с пользовательскими объектами в JavaScript	6	2	4
Тема 14. Классы и наследование в JavaScript	6	2	4
Тема 15. Повышение эффективности работы веб-приложения за счет хранения данных на клиентской стороне	6	2	4
Раздел 3. Приемы, средства и инструменты для разработки клиент-серверных веб-приложений	56	28	28
Тема 16. Особенности функционального программирования и работы с функциями в JavaScript	2	2	-
Тема 17. Организация работы со сложными структурами данных в JavaScript. Работа с коллекциями	2	2	-
Тема 18. Встроенные структуры данных и подходы к их обработке в JavaScript. Использование вспомогательных средств для сокращения и оптимизации кода	6	2	4
Тема 19. Организация модульного подхода при разработке проектов в JavaScript	2	2	-
Тема 20. Основы программирования на стороне сервера в JavaScript. Введение в Node.js. API Node.js	4	2	2
Тема 21. Разработка клиент-серверных приложений на платформе Node.js. Организация работы с файловой системой в Node.js	8	4	4
Тема 22. Асинхронный подход к построению интерактивных пользовательских интерфейсов веб-приложений Ajax	6	2	4
Тема 23. Организация хранения данных в документоориентированных СУБД. Введение в NoSQL. MongoDB	6	2	4
Тема 24. Современные инструменты и технические средства, используемые при разработке клиент-серверных приложений	6	2	4
Тема 25. Перспективные направления разработки интерактивных пользовательских интерфейсов на основе современных библиотек. Библиотека React и ее возможности	6	2	4

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Тема 26. Разработка стабильных высокорезактивных веб-страниц на основе библиотеки React и пути повышения производительности веб-проекта	4	2	2
Тема 27. Варианты оптимизации проектных программных решений. Инструменты и технические средства для оптимизации разработки веб-проекта. Эффективные веб-приложения	2	2	-
Итого:	132	56	76

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. СОЗДАНИЕ, ВЕРТКА И СТИЛЕВОЕ ОФОРМЛЕНИЕ ВЕБ-СТРАНИЦ С РАЗМЕЩЕНИЕМ ГИПЕРТЕКСТОВОЙ ИНФОРМАЦИИ НА ОСНОВЕ СТРУКТУРИРОВАННЫХ ДОКУМЕНТОВ

Тема 1. ГИПЕРТЕКСТОВЫЙ ЯЗЫК HTML. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ХАРАКТЕРИСТИКИ ВЕБ-СТРАНИЦЫ

Понятие веб-страницы. Развитие представления о формате и описании веб-страниц. Правила оформления HTML-кода. Общая структура документа HTML. Языки разметки гипертекста: SGML, HTML, XML, XHTML, DHTML. HTML. Определение версии HTML. Элемент `<!DOCTYPE>`. Информационно значимое содержимое веб-страницы. Браузер и его возможности. Теги. Атрибуты тегов. Понятие веб-элемента. Создание таблиц. Понятие и атрибуты формы. Мультимедиа. Форматы изображений. Audio, video. Введение в SVG. гиперссылки; представление графической информации. Новые теги и возможности HTML5.

Тема 2. ФОРМИРОВАНИЕ СТАТИЧЕСКОГО КОНТЕНТА ВЕБ-СТРАНИЦЫ. РАЗМЕЩЕНИЕ ЭЛЕМЕНТОВ НА СТРАНИЦЕ

Отображение элементов на веб-странице. Правила синтаксиса в HTML-документе. Принцип размещения атрибутов. Стандартные атрибуты. Атрибуты событий. Специальные символы. Использование комментариев. Форматирование текста. Работа со шрифтами. Специальные программные средства для разработки веб-страницы. Применение CAN-VAS. Применение языков MATHML, SMIL.

Тема 3. КАСКАДНЫЕ ТАБЛИЦЫ CSS. ОБЪЯВЛЕНИЕ СТИЛЕЙ

Внешнее оформление стилей веб-страницы. Особенности записи в селекторе тегов. Строковое объявление стилей. Универсальные селекторы, селекторы по ID, классу, атрибутам. Управление фоновым изображением, градиентами, прозрачностью, трансформацией. CSS анимация. Псевдоэлементы. Псевдоклассы. CSS Reset. Методологии CSS. Форматирование контента с помощью CSS.

Тема 4. ВИДЫ СЕЛЕКТОРОВ. РАЗРАБОТКА ДИЗАЙНА ВЕБ-СТРАНИЦ НА ОСНОВЕ КАТЕГОРИЙ СВОЙСТВ И ТИПОВ СЕЛЕКТОРОВ CSS. СТАНДАРТ CSS3 И ЕГО ВОЗМОЖНОСТИ

Использование возможностей стандарта CSS3 для дизайна веб-страниц. Селекторы потомков. Соседние, родственные, сестринские селекторы. Комбинирование селекторов. Препроцессоры. Медиазапросы CSS3. Приоритет селекторов. Кроссбраузерные стили.

Тема 5. СОЗДАНИЕ СТРУКТУРЫ ГИПЕРТЕКСТОВОГО ДОКУМЕНТА НА ОСНОВЕ HTML-РАЗМЕТКИ И БЛОЧНОЙ ВЕРСТКИ МАКЕТОВ ВЕБ-СТРАНИЦ

Структура элемента в блочной верстке. Стилиевые решения для блока. Варианты блочных моделей. Рамка, поля (контуры) и отступы. Форматирование блоков: свойства блоков, позиционирование, обтекание. Семантическая верстка (header, footer, aside и др.). Макеты верстки (фиксированный, статический, эластичный, адаптивный, комбинированный): их достоинства и недостатки. Инструменты для создания макетов веб-страниц.

Тема 6. СОВРЕМЕННЫЕ ПОДХОДЫ К СИСТЕМЕ КОМПАНОВКИ ЭЛЕМЕНТОВ НА ВЕБ-СТРАНИЦЕ И СТИЛИЗАЦИИ РАЗМЕЩЕНИЯ ВЕБ-ЭЛЕМЕНТОВ

Технологии размещения веб-элементов на основе гибких макетов CSS Flexbox, Grid Layout, систем компоновки и актуальных подходов к стилизации веб-страниц. Инструменты для быстрого создания гибких, двумерных и адаптивных макетов (CSS Flex, Grid Layout Generators TailwindCSS, Twitter Bootstrap, Foundation и др.).

Тема 7. ОБЪЕКТНАЯ МОДЕЛЬ ДОКУМЕНТА DOM (DOCUMENT OBJECT MODEL) И ОБЪЕКТНАЯ МОДЕЛЬ БРАУЗЕРА BOM (BROWSER OBJECT MODEL). ВОЗМОЖНОСТЬ ДОСТУПА К ВЕБ-ЭЛЕМЕНТАМ И УЗЛАМ НА СТРАНИЦЕ

Получение доступа к содержимому HTML-, XHTML- и XML-документов, изменение содержимого, структуры и оформления веб-документов. Понятие объектной модели документа, объектной модели браузера. Особенности реализации в разных браузерах. Объекты верхнего уровня. Варианты поиска элементов на странице и получения их значений. Объект document. Методы объекта document для определения элементов структуры DOM. Возможности BOM для управления объектами браузера.

Тема 8. ФОРМИРОВАНИЕ ИНТЕРАКТИВНЫХ ДИНАМИЧЕСКИХ ВЕБ-СТРАНИЦ С ПОМОЩЬЮ ДИНАМИЧЕСКОГО ДОСТУПА К СОДЕРЖИМОМУ ДОКУМЕНТА, ЕГО СТРУКТУРЕ И СТИЛЯМ С ПОМОЩЬЮ DHTML. ОБРАБОТКА СОБЫТИЙ JAVASCRIPT

Понятие и классификация событий. Привязка событий. Динамическое содержание документа. Инструменты интерактивного взаимодействия с пользователями веб-ресурса. Способы перехвата событий. Объект Event. Методы и возможности динамического добавления, удаления, копирования, дублирования, изменения стиля веб-элементов на странице. Использование возможностей DHTML для построения динамических пользовательских интерфейсов. Всплытие событий. Возможность влияния на всплытие событий.

Тема 9. ПРИМЕНЕНИЕ ОБЪЕКТНОЙ МОДЕЛИ ДОКУМЕНТОВ ДЛЯ РАБОТЫ С ФОРМАМИ ВЕДЕНИЯ ДАННЫХ

Определение формы, свойства, атрибуты, стилевое оформление. Создание элементов формы. Размещение и позиционирование веб-элементов в форме ведения данных. Управление фокусом. Отправка данных на сервер и очистка формы. Методы GET и POST. Определение событий формы. Основы работы с формами. Инструменты валидации и проверки правильности заполнения формы. Придание интерактивности форм с помощью динамического HTML.

Раздел 2. ВВЕДЕНИЕ В КЛИЕНТСКИЕ СЕТЕВЫЕ ТЕХНОЛОГИИ. ОСНОВЫ ЯЗЫКА JAVASCRIPT

Тема 10. ВВЕДЕНИЕ В МУЛЬТИПАРАДИГМЕННЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT.

Среда разработки веб-приложений. Поддержка со стороны браузеров. Классификация, место, назначение, возможности и история развития языка JavaScript. Версии, спецификации, редакторы JavaScript. Справочники и документация. Синтаксис языка. Лексемы. Идентификаторы. Ключевые и зарезервированные слова. Типы данных. Особенности типизации. Приведение типов данных. Константы. Переменные. Выражения. Комментарии. Команды ввода и вывода. Этапы разработки сценария. Консольный режим работы. Структура кода. Скрипт и его синтаксис. Соединение с внешним файлом .js. Необходимые инструментальные средства.

Тема 11. ОСНОВНЫЕ КОНСТРУКЦИИ ЯЗЫКА JAVASCRIPT. МАССИВЫ. УПРАВЛЯЮЩИЕ СТРУКТУРЫ И ОРГАНИЗАЦИЯ ЦИКЛОВ. ФУНКЦИИ. ОБРАБОТКА СОБЫТИЙ В JAVASCRIPT

Условные операторы if, if...else. Операторы цикла for, for..in, for..of, while, do..while, break, continue, switch. Массивы JavaScript. Способы создания, заполнения и работы с массивами. Объект Array, его свойства и методы. Понятие функций. Создание функции. Использование параметров. Параметры по умолчанию. Рекурсивные функции. Функциональные выражения. Внутренние функции. Создание сценариев с использованием функций. Локальные и глобальные переменные. Области видимости. Виды событий. Обработчики событий.

Тема 12. ПОНЯТИЕ ОБЪЕКТА JAVASCRIPT. СТАНДАРТНЫЕ ВСТРОЕННЫЕ ОБЪЕКТЫ И ИХ НАЗНАЧЕНИЕ

Объекты JavaScript. Понятие объекта. Тип данных Object. Создание объекта. Свойства, методы объекта и доступы к ним. Встроенные объекты JavaScript. Объекты Document, Window, String и др. Работа со строковыми величинами. Использование стандартных и встроенных объектов при решении различных задач. Объект Function. Использование функциональных выражений. Методы call() и apply() для установки в контекст исполнения (this).

Тема 13. ОРГАНИЗАЦИЯ РАБОТЫ С ПОЛЬЗОВАТЕЛЬСКИМИ ОБЪЕКТАМИ В JAVASCRIPT

Создание пользовательских объектов JavaScript. Копирование объектов и ссылки. Сборка мусора. Массивоподобные объекты. Литерал объекта. Поиск элементов по ключу и значению. Использование Object.keys, values, entries для работы с элементами объектов. Проверка существования свойств с помощью методов hasOwnProperty() и propertyIsEnumerable(). Добавление, изменение и удаление свойств и методов, доступ к ним. Статический метод Object.create(). Оператор «new» для создания экземпляра объекта.

Тема 14. КЛАССЫ И НАСЛЕДОВАНИЕ В JAVASCRIPT

Понятие класса в JavaScript и декларирования классов с помощью ключевого слова class. Конструктор класса. Использование «this» для работы с конструктором класса. Доступ к [[Prototype]] объекта. Организация проверки класса, оператор «instanceof». Разница между __proto__ и Prototype. Добавление свойств и методов в конструктор класса. Наследование классов в языке программирования JavaScript, ключевое слово super, оператор extends, наследование конструкторов. Прототипное наследование. Встроенные прототипы. Методы прототипов. Приватные поля и методы. Статические поля и методы. Свойства и методы доступа.

Тема 15. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ РАБОТЫ ВЕБ-ПРИЛОЖЕНИЯ ЗА СЧЕТ ХРАНЕНИЯ ДАННЫХ НА КЛИЕНТСКОЙ СТОРОНЕ

Концепция хранения данных на стороне клиента и ее преимущества. Использование cookies для хранения информации. Использование HTML5 Application Cache. Локальное и сессионное хранилища Web Storage. Возможности IndexedDB и других актуальных способов хранения данных на стороне клиента.

Раздел 3. ПРИЕМЫ, СРЕДСТВА И ИНСТРУМЕНТЫ ДЛЯ РАЗРАБОТКИ КЛИЕНТ-СЕРВЕРНЫХ ВЕБ-ПРИЛОЖЕНИЙ

Тема 16. ОСОБЕННОСТИ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ И РАБОТЫ С ФУНКЦИЯМИ В JAVASCRIPT

Понятие «чистых функций». Варианты оптимизации и сокращения кода за счет использования стрелочных функций, spread, rest функций, функциональных конструкций (функции-выражения (карринг)). Функции высшего порядка, замыкания, немедленно вызываемые (IIFE), самоопределяющиеся, настраиваемые через bind() и т.д.). Организация планирования вызова функций (setTimeout и setInterval и др.).

Тема 17. ОРГАНИЗАЦИЯ РАБОТЫ СО СЛОЖНЫМИ СТРУКТУРАМИ ДАННЫХ В JAVASCRIPT. РАБОТА С КОЛЛЕКЦИЯМИ

Императивный и декларативный подход к обработке массивов данных. Встроенные методы для обработки элементов массива на декларативной основе. Деструктуризация массивов. Работа с Map, Set, WeakMap и WeakSet. Итератор. Итерируемые объекты. Перебирающие методы.

Тема 18. ВСТРОЕННЫЕ СТРУКТУРЫ ДАННЫХ И ПОДХОДЫ К ИХ ОБРАБОТКЕ В JAVASCRIPT. ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНЫХ СРЕДСТВ ДЛЯ СОКРАЩЕНИЯ И ОПТИМИЗАЦИИ КОДА

Методы для работы с датой и временем. Объект Date, его свойства и методы. Формат JSON, методы toJSON(), JSON.parse(), JSON.stringify(). Применение шаблонных литералов, JavaScript шаблонизаторов (Mustache/Handlebars, EJS, Pug) для структуризации и сокращения программного кода, их применение и назначение.

Тема 19. ОРГАНИЗАЦИЯ МОДУЛЬНОГО ПОДХОДА ПРИ РАЗРАБОТКЕ ПРОЕКТОВ В JAVASCRIPT

Форматы модульной архитектуры в JavaScript. Управление зависимостями: AMD, require.js. Форматы UMD, CommonJS. Нативные модули ECMAScript. Синтаксис выражений для экспорта и импорта. Загрузчики модулей. Принципы организации хранения информации в модулях. Доступ к данным. Динамический импорт. Встроенные и пользовательские модули.

Тема 20. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА СТОРОНЕ СЕРВЕРА В JAVASCRIPT. ВВЕДЕНИЕ В NODE.JS. API NODE.JS.

Основы парадигмы событийно-ориентированного, асинхронного (реактивного) программирования с неблокирующим вводом/выводом. Цикл событий (Event Loop) и библиотека libuv. Стадии Event Loop. Менеджер пакетов npm. Подключение и управление модулями с помощью npm. Встроенные и сторонние модули, потоки. Фреймворк express, его возможности и преимущества. Node.js-библиотеки и модули.

Тема 21. РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНЫХ ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ NODE.JS. ОРГАНИЗАЦИЯ РАБОТЫ С ФАЙЛОВОЙ СИСТЕМОЙ В NODE.JS.

Создание клиент-серверных приложений на платформе Node.js. Организация клиентских запросов к серверу. Инструменты и методы для работы с файлами и файловой системы в Node.js. Создание, перемещение, удаление файлов. Работа с файловыми потоками.

Тема 22. АСИНХРОННЫЙ ПОДХОД К ПОСТРОЕНИЮ ИНТЕРАКТИВНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВЕБ-ПРИЛОЖЕНИЙ AJAX

Использование технологии динамического обращения к серверу без перезагрузки всей страницы. Создание объекта XMLHttpRequest. Динамическая

подгрузка данных на страницу. Отправка сообщений с помощью Ajax. Организация асинхронной работы с помощью функции обратного вызова (callback), async/await. Промисы. Promise API.

Тема 23. ОРГАНИЗАЦИЯ ХРАНЕНИЯ ДАННЫХ В ДОКУМЕНТО-ОРИЕНТИРОВАННЫХ СУБД. ВВЕДЕНИЕ В NOSQL. MONGODB

Введение в NoSQL. Основные категории нереляционных баз данных (БД). Понятие коллекции и документа. Архитектура MongoDB и создание баз данных. Поддержка специальных запросов, индексация, балансировка нагрузки, механизм репликации в MongoDB. Установка, организация хранения данных, основные команды, динамические запросы документов. Использование документо-ориентированной СУБД MongoDB в проектах Node.js. Понятие ODM (Object Document Mapper – объектно-документный отображитель). Использование библиотеки Mongoose для доступа к базе данных из приложений.

Тема 24. СОВРЕМЕННЫЕ ИНСТРУМЕНТЫ И ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ ПРИ РАЗРАБОТКЕ КЛИЕНТ-СЕРВЕРНЫХ ПРИЛОЖЕНИЙ.

Бандлинг (компоновка) модулей. Сборщик модулей webpack. Системы сборки. Планировщик задач. Пакетные менеджеры. Менеджеры задач.

Тема 25. ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ РАЗРАБОТКИ ИНТЕРАКТИВНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ НА ОСНОВЕ СОВРЕМЕННЫХ БИБЛИОТЕК. БИБЛИОТЕКА REACT И ЕЕ ВОЗМОЖНОСТИ

Технология разработки одностраничных SPA (single-page application), многостраничных (MPA) и прогрессивных (PWA) приложений. Использование декларативного дизайна. Настройка окружения. React-компоненты: виды компонентов, компонент с состоянием, компонент с использованием внешних плагинов. Архитектура React.js. Виртуальный DOM (VDOM). Синтаксис JSX. Рендеринг ReactDOM. Метод ReactDOM.render() для отображения React- или DOM-элемента. Состояние (state).

Тема 26. РАЗРАБОТКА СТАБИЛЬНЫХ ВЫСОКОРЕАКТИВНЫХ ВЕБ-СТРАНИЦ НА ОСНОВЕ БИБЛИОТЕКИ REACT И ПУТИ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ВЕБ-ПРОЕКТА

Сборка компонентов. Классы компонентов и функциональные компоненты. Обмен данными между компонентами. Изменение компонента React на основе его состояния. Обработка событий. Асинхронный метод setState. Свойства (props). Ссылки (refs). Повышение производительности и скорости загрузки страниц за счет использования virtual DOM и «ленивой» загрузки, библиотеки готовых компонентов. Хуки. Браузерные инструменты React-разработчика.

Тема 27. ВАРИАНТЫ ОПТИМИЗАЦИИ ПРОЕКТНЫХ РЕШЕНИЙ.
ИНСТРУМЕНТЫ И ТЕХНИЧЕСКИЕ СРЕДСТВА ДЛЯ ОПТИМИЗАЦИИ
РАЗРАБОТКИ ВЕБ-ПРОЕКТА. ЭФФЕКТИВНЫЕ ВЕБ-ПРИЛОЖЕНИЯ

Пути повышения производительности веб-приложения. Оптимизация скорости разработки. Подходы к client-side оптимизации. Минификация и оптимизация HTML, CSS и JavaScript кода. Оптимизация картинок, шрифта и т.д. Применение кэширования и оптимизация доставки контента. Минимизация HTTP-запросов. Использование ajax-запросов и анализа передачи данных с сервера. Инструменты для оценки производительности сайта.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ЛИТЕРАТУРА

ОСНОВНАЯ

1. Хавербеке, М. Выразительный JavaScript : современное веб-программирование / М. Хавербеке. – 3-е изд. – Санкт-Петербург : Питер, 2022. – 480 с.
2. Дронов, В. А. HTML и CSS: 25 уроков для начинающих / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2023. – 400 с.
3. Кириченко, А. В. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков / А. В. Кириченко, А. П. Никольский, Е. В. Дубовик. – Санкт-Петербург : Наука и Техника, 2021. – 432 с.
4. Сидельников, Г. Наглядный CSS / Г. Сидельников. – Санкт-Петербург : Питер, 2022. – 224 с.
5. Грант, К. CSS для профи / К. Грант. – Санкт-Петербург : Питер, 2021. – 496 с.
6. Янг, А. Node.js в действии / А. Янг, Б. Мек, М. Кантелон. – 2-е изд. – Санкт-Петербург : Питер, 2018. – 432 с.
7. Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон. – Санкт-Петербург : Питер, 2019. – 640 с.
8. Фрисби, М. JavaScript для профессиональных веб-разработчиков / М. Фрисби. – 4-е изд. – Санкт-Петербург : Питер, 2022. – 1168 с.
9. Аквино, К. Front-end. Клиентская разработка для профессионалов : Node.js, ES6, REST / К. Аквино, Т. Ганди. – Санкт-Петербург : Питер, 2017. – 408 с.
10. Титтел, Э. HTML5 и CSS3 для чайников / Э. Титтел, К. Минник ; пер. с англ. – Москва ; Санкт-Петербург : Диалектика, 2020. – 400 с.
11. Дронов, В. А. React 17. Разработка веб-приложений на JavaScript / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2022. – 384 с.
12. Макфарланд, Д. Большая книга CSS3 / Д. Макфарланд ; пер. с англ. Н. Вильчинского. – 3-е изд. – Санкт-Петербург : Питер, 2015. – 608 с.

ДОПОЛНИТЕЛЬНАЯ

13. Столлингс, В. Компьютерные сети, протоколы и технологии Интернета / В. Столлингс. – Санкт-Петербург: БХВ-Петербург, 2005. – 832 с.
14. Дронов, В. А. Java Script. 20 уроков для начинающих / В. А. Дронов. – Санкт-Петербург: БХВ-Петербург, 2022. – 352 с.
15. Ллойд, Й. Создай свой веб-сайт с помощью HTML и CSS / Й. Ллойд. – Санкт-Петербург: Питер, 2013. – 416 с.
16. Холмогоров, В. Основы Web-мастерства: учебный курс / В. Холмогоров. – Санкт-Петербург: Питер, 2011. – 352 с.
17. Дакетт, Дж. HTML и CSS. Разработка и дизайн веб-сайтов / Дж. Дакетт. – Санкт-Петербург: Питер, 2013. – 480 с.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- написание рефератов и создание материалов презентаций;
- составление тестов и эталонов ответов к ним;
- подготовка к контрольной работе или коллоквиуму;
- участие обучающихся в научно-исследовательской и методической работе, проводимой на кафедре;
- участие в конкурсах студенческих работ и студенческих конференциях.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ

Примерным учебным планом по специальности 6-05-0611-04 «Электронная экономика» в качестве формы промежуточной аттестации по учебной дисциплине «Веб-технологии и программирование» рекомендуется зачет и экзамен. Оценка учебных достижений обучающихся производится по системе «зачтено/не зачтено» и десятибалльной шкале.

Для текущего контроля по учебной дисциплине и диагностики компетенций могут использоваться следующие формы:

- устный опрос;
- защита лабораторной работы;
- собеседование;
- защита индивидуальной практической работы;
- защита контрольной работы;
- коллоквиум.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

Основные рекомендуемые методы (технологии) обучения, отвечающие целям и задачам учебной дисциплины:

- устное изложение учебного материала на лекциях, сопровождаемое показом и демонстрацией слайдов, видеоматериалов;
- компьютерное обучение;
- элементы проблемного обучения (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемые на лекционных занятиях;
- элементы учебно-исследовательской деятельности, реализуемые на лабораторных занятиях.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Создание веб-страницы с использованием стилового оформления (определенный шрифт, параметры и атрибуты веб-элементов и т.д.) с помощью каскадной таблицы стилей CSS и возможностей CSS3.
2. Разработка веб-проекта с интерактивными динамическими веб-страницами с использованием динамического доступа к содержимому документа, его структуре и стилям с помощью DHTML.
3. Создание проекта с использованием сценариев на основе методов и функций стандарта ECMA 6, структуры данных Map и/или Set, для поиска, фильтрации и обработки данных.
4. Создание веб-проекта с использованием сценариев динамической подгрузки данных и классов JavaScript.
5. Разработка динамических веб-проектов с использованием сервера Node.js на основе модульной системы.
6. Разработка клиент-серверного приложения на основе сервера Node.js и документо-ориентированной базы данных Mongo DB.
7. Моделирование веб-проекта с использованием асинхронной подгрузки данных и обработки событий (элементы технологии Ajax) на основе платформы Node.js и асинхронных методов создания динамических интерактивных веб-страниц.
8. Разработка веб-проекта на основе сервера Node.js и библиотеки React.js с использованием бандлинга.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ

(необходимого оборудования, наглядных пособий и др.)

1. Операционная система (например, Windows).
2. Браузеры Google Chrome, Яндекс, Firefox, Safari, Opera.
3. Интегрированная среда разработки Visual Studio Code или редакторы: Sublime Text, Atom, WebStorm, notepad++.
4. Программная платформа Node.js.