

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ  
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО  
ОБРАЗОВАНИЯ РЕСПУБЛИКИ КОМИ  
«РЕСПУБЛИКАНСКИЙ ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ»

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

**технической направленности**

**«ОСНОВЫ ПРОГРАММИРОВАНИЯ САЙТОВ  
И ВЕБ-ПРИЛОЖЕНИЙ»**

Возраст детей: 14-17 лет

Срок реализации: 1 год

Составитель:  
**Бушенев Игорь Владимирович,**  
педагог дополнительного образования

Сыктывкар  
2023

## Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования сайтов и веб-приложений» (далее - программа) составлена в соответствии с действующими нормативными правовыми документами.

*Направленность* программы – техническая. Программа предназначена для получения школьниками дополнительного образования в области современных информационных технологий.

*Новизна программы.*

В основу программы положены новые технологии образования, учитывающие запросы учащихся и потребности современного общества в подготовке будущих квалифицированных инженерных кадров. Программа имеет профессионально – технический подход (освоение основ профессий технической направленности, а также связанные с цифровыми технологиями). В ходе обучения учащиеся выполняют различные задачи, которые позволяют развить критическое мышление, способность анализировать ситуацию, учиться ставить перед собой цели и задачи, расставлять акценты, прогнозировать ожидаемый результат.

*Актуальность программы.*

Развитие сети Интернет предъявляет все большие требования к знанию учащихся в области Интернет-технологий. Работа с информацией стала отдельной специальностью, остро востребованной на рынке труда. В настоящее время компьютерная грамотность является неотъемлемой частью профессиональной жизни человека во всех сферах деятельности. При этом от специалиста, требуется уровень уверенного и даже продвинутого пользователя, владеющего приемами программирования, умеющего работать в сети Интернет, пользоваться сетевыми и коммуникационными сервисами, эффективно обрабатывать все виды информации. Одной из составляющих данной области является веб-разработка. Именно на веб-разработчиков возложена обязанность создания удобного, простого и понятного способа навигации в сети Интернет. Реалии сегодняшнего дня таковы, что любой желающий может создать свой собственный веб-сайт и разместить его абсолютно бесплатно, в сети Интернет.

Основная идея программы заключается в использовании разноуровневых дифференцированных заданий, расположенных по принципу «от простого к сложному», что позволяет стимулировать мотивацию учащихся к освоению данной программы и учитывать индивидуальные особенности каждого учащегося. В этом заключается *педагогическая целесообразность* программы.

Своеобразие программы заключается в ее направленности на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования. А также проработка базовых тем на

уровне, достаточном для решения реальных задач в профессии веб-разработчика.

*Адресат программы:* учащиеся в возрасте 14 - 17 лет. Дети раннего подросткового возраста отличаются высокой эффективностью мышления. Высокая степень мотивации, обусловленная потребностью в общении и стремлением к самоутверждению, позволяет учащимся ставить перед собой амбициозные цели.

*Вид программы по уровню освоения:* программа рассчитана на базовый уровень освоения знаний и навыков в области веб-разработки и программирования.

*Объем программы и режим занятий*

Год обучения	Количество детей в группе	Продолжительность одного занятия	Всего часов в неделю	Количество часов в год
I	12	40 минут	6	216

*Формы организации образовательного процесса:*

Групповые, в основе процесса деятельности – индивидуальный подход к учащемуся.

Программой предусмотрено проведение комбинированных занятий. Занятия состоят из теоретической и практической частей, с преобладанием практической части. При проведении занятий используются следующие формы работы:

На этапе изучения нового материала – лекции, объяснение, рассказ, демонстрация;

На этапе закрепления изученного материала - беседы, дискуссии, лабораторно-практическая работа, дидактическая или педагогическая игра;

На этапе повторения изученного материала – наблюдение, устный контроль (опрос, игра);

На этапе проверки полученных знаний – тестирование, выполнение дополнительных заданий, публичное выступление с демонстрацией результатов работы.

Самостоятельная работа с учащимися – планируемая учебная работа учащихся, выполняемая во внеаудиторное время по заданию и при методическом руководстве и консультативной помощи педагога, но без его непосредственного участия с использованием электронного обучения и/или дистанционных образовательных технологий в части реализации программы:

- в рамках онлайн занятий посредством онлайн-платформ педагог представляет теоретический материал по теме;

- в офлайн режиме посредством социальных сетей и мессенджеров учащимся передается видео, презентационный материал с инструкцией выполнения заданий, мастер-классы и др.

- индивидуальные консультации в онлайн-чатах.

В качестве форм и методов контроля самостоятельной работы, учащихся могут быть использованы фронтальные опросы на практических занятиях, зачеты, тестирование, творческий продукт учащегося и др.

*Методическое обеспечение программы.* Возможные формы проведения занятий: теоретическое занятие, практическое занятие, комбинированные занятия, занятие-соревнование, экскурсия, консультация, выставка. Рекомендуемые методы проведения занятий: кейс-метод, проектная деятельность, дарт скаутинг. При проведении занятий используются следующие формы работы с учащимися: самостоятельная работа, исследовательский метод обучения, лекция-диалог с использованием метода «перевернутый класс» – когда учащимся предлагается к следующему занятию ознакомиться с материалами (в том числе найденными самостоятельно) на определенную тему для обсуждения в формате диалога на предстоящем занятии, workshop и tutorial (практическое занятие – hardskills), проблемное изложение материала, когда ставится задача, позволяющая решить определенный этап процесса обучения и перейти на новую ступень обучения.

*Воспитательная работа* с учащимися реализуется в соответствии с Программой воспитания ГАУДО РК «Республиканский центр дополнительного образования» и плана работы на учебный год Центр цифрового образования детей «IT-куб».

### **Цель и задачи программы**

**Цель** – формирование у учащихся компетенций в области веб-разработки, информационных технологий и программирования, способствующих их профессиональному самоопределению.

#### ***Задачи:***

##### ***Образовательные:***

1. знакомство с основными принципами и методами веб-разработки, включая HTML, CSS, JavaScript и других технологий, принципами работы веб-серверов и баз данных, формирование способности интегрировать их с веб-приложениями;
2. - создание структуры сайтов и разработка их дизайн с использованием современных инструментов и технологий, разработка адаптивного и кроссбраузерного дизайна веб-сайтов, учитывая различные устройства и разрешения экранов, обучение работать с системами контроля версий, проектировать и разрабатывать веб-приложения с использованием серверных языков программирования;
3. - формирование навыков программирования на языках HTML, CSS и JavaScript, а также опыт работы с фреймворками и библиотеками.

##### ***Развивающие:***

1. формирование умения использовать компьютер как инструмент познания и саморазвития в техническом направлении;
2. формирование и развитие творческих способностей учащихся;
3. формирование эстетического и художественного вкуса;

4. развитие образного мышления, внимания, фантазии;
5. развитие коммуникативных умений и навыков, обеспечивающих совместную проектную деятельность учащихся.

*Воспитательные:*

1. развитие воображения и фантазии при выполнении творческих работ;
2. воспитание эстетического отношения к действительности, трудолюбия, аккуратности, усидчивости, терпения, умения довести начатое дело до конца, взаимопомощи при выполнении работы;
3. привитие основ культуры труда;
4. воспитание умения и навыков межличностного сотрудничества.

### Содержание программы Учебный план

№ п/п	Название раздела	Всего часов	Количество часов		Формы аттестации/ко нтроля
			Теория	Практика	
1.	Основы верстки веб-страниц. HTML и CSS	45	14	31	Практическое задание
2.	Продвинутая верстка веб-страниц	27	8	19	Практическое задание
3.	Основы JavaScript. Программирование на стороне клиента.	21	7	14	Практическое задание
4.	JavaScript в браузере	27	9	18	Практическое задание
5.	Разработка динамических веб-приложений	15	4	11	Практическое задание
6.	Серверная часть сайтов и веб-приложений (backend PHP+SQL)	48	15	33	Практическое задание
7.	Проектная деятельность	21	3	18	Проектная работа
8.	Повторение изученного материала, обобщение, подготовка к аттестации.	3	3	-	Практическое задание
9.	Презентация и защита проекта.	9	3	6	Защита проекта, тестирование уровня знаний.
<b>ИТОГО:</b>		<b>216</b>	<b>66</b>	<b>150</b>	

## Содержание учебного плана

### **Раздел 1. Основы верстки веб-страниц. HTML и CSS.**

#### **Тема 1.1. Введение. Техника безопасности. Правила поведения на занятиях. Входной контроль.**

*Теория: вводная диагностика уровня знаний, умений, навыков по математике и информатике в форме диалога. Инструктаж по ТБ. Содержание, цели и задачи программы.*

#### **Тема 1.2. Знакомство с общими понятиями о веб-разработке. Настройка рабочего окружения.**

*Теория: как устроен интернет. Как происходит обмен данными в сети. Что такое «клиент-серверное взаимодействие». Какие технологии необходимо знать для создания веб-сайтов.*

#### **Тема 1.3. Настройка рабочего окружения. Введение в HTML.**

*Теория: обзор инструментов для работы. Знакомство с языком разметки веб-страниц. Структура HTML-документа. Синтаксис языка HTML. Понятие тега в HTML и их разновидности.*

*Практика: установка и настройка редактора кода Visual Studio Code. Первый созданный веб-макет.*

#### **Тема 1.4. Вложенные теги. Атрибуты тегов. Блочные и строчные теги.**

*Теория: понятие вложенных тегов. Что такое - атрибуты тегов. Разновидности тегов: inline и block теги.*

*Практика: применение различных HTML-тегов на практике.*

#### **Тема 1.5. Работа с изображениями в HTML.**

*Теория: создание и вставка изображений в веб-страницу. Работа с атрибутами изображений.*

*Практика: верстка страниц с использованием изображений.*

#### **Тема 1.6. Верстка списков и таблиц в HTML.**

*Теория: изучение новых тегов для построения конструкций списков и таблиц. Практика создания веб-страниц со списками и таблицами.*

*Практика: создание веб-страниц со списками и таблицами.*

#### **Тема 1.7. Создание веб-форм.**

*Теория: понятие веб-формы и управляющих тегов. Виды управляющих тегов.*

*Практика: верстка различных веб-форм.*

#### **Тема 1.8. Обзор HTML5-тегов.**

*Теория: обзор возможностей HTML5-тегов.*

*Практика: применение HTML5-тегов.*

#### **Тема 1.9. Практика создания полноценной HTML-разметки.**

*Практика: создание полноценной HTML-разметки.*

#### **Тема 1.10. Знакомство с каскадной таблицей стилей CSS. Применение простейших свойств стилизации HTML-разметки.**

*Теория: понятия каскадной таблицы стилей. Способы подключения стилей к html-странице. Синтаксис языка CSS.*

*Практика: применение различных CSS-свойств.*

**Тема 1.11. Селекторы или способы обращения к элементам страницы для задания стилей. Комбинированные селекторы.**

*Теория: понятие селектора класса, тега, атрибута, идентификатора. Вложенные селекторы, комбинированные селекторы.*

*Практика: применение различных селекторов для создания стилей.*

**Тема 1.12. Блочная модель CSS.**

*Теория: изучение блочной модели веб-страницы. Задание границ, отступов и размеров блокам.*

*Практика: применение CSS-свойств для построения нужной блочной модели.*

**Тема 1.13. Позиционирование в CSS.**

*Теория: основы позиционирования в верстке. Понятие потока элементов на странице. Абсолютное и относительное позиционирование в CSS.*

*Практика: применение позиционирования в верстке.*

**Тема 1.14. Псевдо-классы и псевдо-элементы в CSS.**

*Теория: понятие псевдоклассов и псевдоэлементов в CSS. Стилизация состояний элементов.*

*Практика: стилизация состояний.*

**Тема 1.15. Практика создания веб-страниц со структурой и оформлением.**

*Практика: создание веб-страниц со структурой и оформлением.*

**Тема 1.16. Переходы и анимация в CSS.**

*Теория: примеры создания анимации при помощи CSS.*

*Практика: создание CSS-свойств для задания анимации элементам страницы.*

**Раздел 2. Продвинутая верстка веб-страниц.**

**Тема 2.1. Основы верстки по макету.**

*Теория: знакомства с инструментами для работы с готовыми макетами сайтов.*

*Практика: верстка по макету.*

**Тема 2.2. Создание структуры веб-страниц.**

*Теория: введение в продвинутую верстку. Понятие валидности и семантики веб-страницы. Понятия шапки, тела и подвала страницы.*

*Практика: создание валидной и семантически верной верстки.*

**Тема 2.3. Технология flexbox-верстки.**

*Теория: обзор технологии flexbox для построения верстки страницы.*

*Практика: применение flexbox.*

**Тема 2.4. Введение в резиновую и адаптивную верстку. Медиа-запросы в CSS.**

*Теория: понятие резиновой и адаптивной верстки. Применение относительных единиц измерения для вёрстки. Медиа-запросы в CSS как способ создания адаптивных страниц.*

*Практика: создание резиновой и адаптивной верстки.*

### **Тема 2.5. Верстка резиновых и адаптивных изображений.**

*Теория: применение принципов адаптивности к верстке изображений.*

*Создание резиновых изображений.*

*Практика: верстка резиновых и адаптивных изображений.*

### **Тема 2.6. Верстка резиновых и адаптивных веб-страниц.**

*Теория: применение принципов адаптивности к верстке полноценных страниц. Создание резиновых страниц.*

*Практика: верстка резиновых и адаптивных страниц.*

### **Тема 2.7. Знакомство с фреймворками для быстрого построения адаптивных страниц.**

*Теория: понятие фреймворка. Фреймворк как инструмент для быстрого построения веб-страниц. Обзор популярных CSS-фреймворков. Введение во Фреймворк Bootstrap.*

*Практика: применение фреймворков для верстки.*

### **Тема 2.8. Понятие сетки для верстки. Применение фреймворков в работе.**

*Теория: понятие сетки и построения страниц по сетке.*

*Практика: применение фреймворка Bootstrap для верстки страниц по сетке.*

### **Тема 2.9. Практика создания адаптивных страниц с использованием фреймворков.**

*Практика: создание адаптивных страниц с использованием фреймворков.*

## **Раздел 3. Основы JavaScript. Программирование на стороне клиента.**

### **Тема 3.1. Введение в программирование на JavaScript.**

*Теория: программирование на стороне клиента. Знакомство с языком программирования JavaScript. Ввод-вывод в JavaScript. Базовый синтаксис языка. Способы подключения JavaScript к веб-странице.*

*Практика: программирование на стороне клиента.*

### **Тема 3.2. Переменные. Типы данных. Арифметика.**

*Теория: погружение в язык JavaScript. Способы создания переменных. Базовые типы данных в JavaScript. Применение арифметических операторов.*

*Практика: программирование на JavaScript.*

### **Тема 3.3. Алгоритмы ветвления.**

*Теория: понятие линейности кода. Понятия условного оператора и условного выражения. Простые и составные условия.*

*Практика: создание программ с применением алгоритмов ветвления.*

### **Тема 3.4. Циклические алгоритмы.**

*Теория: понятие цикла. Обзор и примеры применения циклических конструкций в JavaScript.*

*Практика: создание программ с применением циклических алгоритмов.*



### **Тема 3.5. Массивы JavaScript.**

*Теория: создание и применение массивов в JavaScript. Циклический перебор элементов массива. Методы массивов.*

*Практика: применение массивов в JavaScript.*

### **Тема 3.6. Функции в JavaScript.**

*Теория: понятие функции как подпрограммы. Синтаксис создания функций в JavaScript. Область видимости переменных. Стрелочные функции. Функции высшего порядка.*

*Практика: применение функций в JavaScript.*

### **Тема 3.7. Объекты в JavaScript.**

*Теория: понятие объекта в JavaScript. Задание свойств объекту. Понятие ссылки на объект.*

*Практика: создание объектов в JavaScript.*

## **Раздел 4. JavaScript в браузере.**

### **Тема 4.1. Введение в DOM. Взаимодействие JavaScript-программы с веб-страницей. Получение и изменение элементов.**

*Теория: понятие DOM (Document Object Model). Способы получения элементов страницы в JavaScript. Понятие узла страницы.*

*Практика: простейшие манипуляции с разметкой через JavaScript.*

### **Тема 4.2. DOM. Навигация по дереву элементов страницы при помощи JavaScript.**

*Теория: способы навигации по странице через JavaScript. Считывание и изменение стилей и атрибутов узлов.*

*Практика: изменение стилей и атрибутов узлов через JavaScript.*

### **Тема 4.3. Изменение структуры HTML-документа при помощи JavaScript.**

*Теория: понятие события. Обработка событий на странице. Предотвращение действий по умолчанию.*

*Практика: обработка событий на веб-странице.*

### **Тема 4.4. События в JavaScript.**

*Теория: Вводная лекция о "облачных технологиях".*

*Практика: Регистрируемся на популярных ресурсах: Яндекс и Google.*

### **Тема 4.5. Работа с HTML-формами через JavaScript.**

*Теория: взаимодействие с формами через JavaScript.*

*Практика: создание скриптов для валидации и иных манипуляций с формами.*

### **Тема 4.6. Асинхронные запросы. Работа с API.**

*Теория: введение в асинхронное программирование и API. Выполнение асинхронных запросов через JavaScript. Знакомство с JSON.*

*Практика: выполнение асинхронных запросов через JavaScript.*

### **Тема 4.7. Хранение состояния и данных на клиенте.**

*Теория: обзор вариантов хранения данных на стороне пользователя и почему это полезно. Основы работы с Cookies и Web Storage.*

*Практика: создание хранилищ данных на стороне клиента.*

#### **Тема 4.8. Подключение сторонних модулей и библиотек.**

*Теория: обзор различных сторонних модулей и библиотек для JavaScript.*

#### **Тема 4.9. Реализация полноценной frontend части веб-сайта.**

*Практика: реализация полноценной frontend части веб-сайта.*

### **Раздел 5. Разработка динамических веб-приложений.**

#### **Тема 5.1. Введение в тему создания SPA приложений.**

*Теория: понятие Одностраничных (SPA) и многостраничных (PWA) веб-приложений. Основные концепции создания SPA.*

*Практика: создание простейшего SPA.*

#### **Тема 5.2. Работа с Canvas. Создание и управление холстом для игровых веб-приложений.**

*Теория: веб игра - как пример SPA приложения. Основы создания веб-игр. Обзор необходимых инструментов.*

*Практика: создание игрового холста.*

#### **Тема 5.3. Реализация Drag-and-drop технологии.**

*Теория: понятие и способы реализации Drag-and-drop (Перетащить и отпустить).*

*Практика: реализация Drag-and-drop на веб-странице.*

#### **Тема 5.4. Изучение и реализация основных игровых механик.**

*Теория: обзор основных игровых механик.*

*Практика: Практика реализации игровых механик.*

#### **Тема 5.5. Практика создания простейших веб-игр.**

*Практика: создание простейших веб-игр.*

### **Раздел 6. Серверная часть сайтов и веб-приложений (backend PHP+SQL).**

#### **Тема 6.1. Введение в разработку backend части веб-проектов.**

*Теория: понятие серверной части веб-проекта(backend). Для чего необходима эта часть. Обзор применяемых технологий.*

*Практика: подготовка рабочей среды. Развертывание локального сервера.*

#### **Тема 6.2. Введение в язык PHP.**

*Теория: язык программирования PHP. Обзор возможностей языка для backend разработки. Базовый синтаксис языка. Переменные. Арифметика.*

*Практика: создание первых программ на PHP.*

#### **Тема 6.3. Условные операторы и алгоритмы ветвления.**

*Теория: условный оператор и условные выражения в PHP.*

*Практика: создание программ с применением алгоритмов ветвления.*

#### **Тема 6.4. Циклические алгоритмы.**

*Теория: разновидности и примеры создания циклов в PHP.*

*Практика: создание программ с применением циклических алгоритмов.*

#### **Тема 6.5. Массивы.**

*Теория: обзор создания и работы с массивами в PHP.*

*Практика: написание программ с применением массивов.*

### **Тема 6.6. Ассоциативные и многомерные массивы.**

*Теория: понятие ассоциативного и многомерного массива в PHP.*

*Практика: написание программ с применением ассоциативных и многомерных массивов.*

### **Тема 6.7. Функции.**

*Теория: создание функций в PHP. Применение встроенных функций.*

*Практика: создание программ с применением функций.*

### **Тема 6.8. Взаимодействие PHP с веб-страницей. Работа с веб-формами.**

*Теория: принцип работы HTTP протокола. Понятие заголовка и тела запроса. Работа с html-формами через PHP.*

*Практика: обработка веб-форм через PHP.*

### **Тема 6.9. Работа с sessions и cookies. Хранение данных на стороне сервера.**

*Теория: принцип работы сессий и куки на стороне сервера.*

*Практика: реализация хранения данных на стороне сервера через PHP.*

### **Тема 6.10. Введение в базы данных. Язык запросов SQL.**

*Теория: понятие баз данных. Обзор разновидностей БД и их возможностей. Введение в язык запросов SQL.*

*Практика: настройка рабочей среды для работы с БД.*

### **Тема 6.11. Создание таблиц. Основные запросы к базе данных.**

*Теория: примеры работы с БД. Создание и удаление таблиц. Понятие поля и значения в таблице. Обновление, добавление и удаление полей.*

*Практика: создание и наполнение БД.*

### **Тема 6.12. Проектирование связанных таблиц.**

*Теория: базы данных с несколькими таблицами. Принципы проектирования сложных БД. Способы взятия многотабличной выборки.*

*Практика: работа с многотабличными БД.*

### **Тема 6.13. Введение в объектно-ориентированное программирование в PHP.**

*Теория: понятие ООП, класса и объекта. Создание собственных классов. Инкапсуляция.*

*Практика: создание классов и объектов в PHP.*

### **Тема 6.14. Наследование и полиморфизм.**

*Теория: понятие наследования и полиморфизма в ООП.*

*Практика: создание иерархии классов.*

### **Тема 6.15. Проектирование классов.**

*Теория: подходы к проектированию классов.*

*Практика: проектирование классов.*

### **Тема 6.16. Практика создания full stack веб-приложения.**

*Практика: создание full stack веб-приложения.*

## **Раздел 7. Проектная деятельность.**

*Теория: Генерация проектных идей, выявление проблемы, анализ существующих технических решений выявленных проблем, формы проекта, жизненный цикл проекта, командный подход к реализации проекта, формы защиты проекта.*

*Практика: Разработка идеи проекта, выбор методов его реализации, применение технологий при реализации проекта, отбор технических инструментов для реализации проекта, разработка уровней проекта, выявление и устранение проблем и технических ошибок, итоговое тестирование проекта и подготовка к защите.*

## **Раздел 8. Повторение изученного материала, обобщение, подготовка к итоговому контролю.**

### **Тема 8. Обобщение изученного материала.**

*Теория: Повторение изученного ранее материала, выполнение тренировочных тестов, ознакомление с формами контроля, критериями оценивания, рефлексия по результатам обучения за год.*

## **Раздел 9. Презентация и защита проекта. Итоговый контроль.**

### **Тема 9. Презентация и защита проекта.**

*Теория: Диагностика уровня сформированности понятийного аппарата учащихся о технологиях программирования на языке Python.*

*Практика: Представление своего проекта (указывается спектр применения, сложности при создании, запуске и их пути решений) и его защита.*

Календарно-тематический план (Приложение 1)

## **Планируемые результаты**

### *Личностные результаты:*

- воспитаны эстетическое отношение к действительности, трудолюбие, аккуратность, усидчивость, терпение, умение довести начатое дело до конца, взаимопомощи при выполнении работы;
- привиты основ культуры труда;
- воспитано умение и навыки межличностного сотрудничества.

### *Метапредметные результаты:*

- сформированы умения использовать компьютер как инструмент познания и саморазвития в техническом направлении;
- сформированы и развиты творческие способности учащихся;
- сформированы эстетический и художественный вкус;
- развито образное мышление, воображение, внимание, фантазия;
- развиты коммуникативные умения и навыки, обеспечивающие совместную проектную деятельность учащихся.

### *Предметные результаты:*

- учащиеся будут знать основные принципы и методы веб-разработки, включая HTML, CSS, JavaScript и другие технологии, принципы работы веб-серверов и баз данных, будут иметь способность интегрировать их с веб-приложениями;

- учащиеся будут уметь создавать структуры сайтов и разрабатывать их дизайн с использованием современных инструментов и технологий, разрабатывать адаптивный и кроссбраузерный дизайн веб-сайтов, учитывая различные устройства и разрешения экранов, работать с системами контроля версий, проектировать и разрабатывать веб-приложения с использованием серверных языков программирования;

- учащиеся будут применять навыки программирования на языках HTML, CSS и JavaScript, а также опыт работы с фреймворками и библиотеками.

Итоги реализации программы могут подводиться в следующих формах: мини-конференция по защите проектов, внутригрупповой конкурс (соревнование), презентация (самопрезентация) проектов учащихся.

### **Условия реализации программы**

#### *Материально-техническое обеспечение:*

Для успешного освоения программы необходимо следующее: учебный кабинет, оборудованный рабочими местами. Кабинет должен иметь хорошее естественное и искусственное освещение, соответствующее санитарно-эпидемиологическим нормативам для данного вида деятельности: учебную доску, столы, стулья.

№ п/п	Наименование	Количество	Назначение/краткое описание функционала оборудования
1	Компьютерное оборудование		
1.1	Компьютер или ноутбук	12	Для основной учебной работы
1.2	Мышь	12	Для основной учебной работы
1.3	Компьютер педагога	1	Для демонстрации учебного материала и выполнения текущих задач
1.4	Мышь	1	Для работы педагога
1.5	Клавиатура	1	Для работы педагога
1.6	Локальная сеть с доступом в интернет.	1	Для работы педагога и учеников.
2	Презентационное оборудование		

2.1	Монитор для презентаций	1	Для теоретических блоков и презентаций
2.2	Напольная мобильная стойка для интерактивных досок с площадкой для крепления проекторов к стойке	1	Для работы с интерактивной доской
3	Программное обеспечение		
3.1	Среда разработки Visual Studio Code	13	Для изучения и использования в практической работе
3.2	Программный комплекс для развертывания локального сервера OpenServer, XAMPP	13	Для изучения и использования в практической работе
3.3	Программы для работы с макетами: Adobe Photoshop, Adobe XD, Figma, Avocode или др.	1	Для изучения и использования в практической работе
4	Мебель		
4.1	КРЕСЛО ДЕТСКОЕ	15	Размещение учеников в учебном кабинете
4.2	Кресло педагога	1	Для рабочего места педагога
4.3	Стол Универсальный	3	Для организации работы в группе
4.4	Стол ученический, с регулируемыми по высоте опорами	6	Размещение учеников в учебном кабинете
4.5	Шкаф для документов	2	Для организации рабочего пространства

#### *Кадровое обеспечение:*

Кадровое обеспечение при реализации программы включает привлечение специалистов с высшим или средним профессиональным образованием в области информационных технологий, программирования и веб-разработки. Педагогические работники должны иметь опыт работы в соответствующей сфере и регулярно повышать свою квалификацию. Также возможно привлечение специалистов-практиков из IT-компаний для проведения мастер-классов, семинаров и проектной деятельности.

#### **Формы контроля и аттестации**

#### *Мониторинг образовательных результатов.*

Система отслеживания, контроля и оценки результатов обучения по данной программе имеет три основных критерия:

1. Надежность знаний и умений
2. Сформированность личностных качеств
3. Готовность к продолжению обучения в данной области, к соревновательной и публичной деятельности.

Способы определения результативности реализации программы и формы подведения итогов реализации программы.

В процессе обучения проводятся разные виды контроля результативности усвоения программного материала.

Текущий контроль проводится на занятиях в виде наблюдения за успехами каждого обучающегося, процессом формирования компетенций. Текущий контроль успеваемости носит безотметочный характер и служит для определения педагогических приемов и методов для индивидуального подхода к каждому обучающемуся, корректировки плана работы с группой.

Промежуточный контроль проводится по окончании изучения каждой темы в виде конкурсов или представления практических результатов выполнения заданий. Конкретные проверочные задания разрабатывает педагог с учетом возможности проведения анализа процесса формирования компетенций. Промежуточный контроль проводится в виде педагогического анализа результатов тестирования, зачетов, опросов, выполнения учащимися диагностических заданий, участия учащихся в мероприятиях (викторинах, соревнованиях), активности учащихся на занятиях и т.п.

Итоговый контроль проводится в виде педагогического анализа результатов выполнения обучающимися диагностических заданий, участия учащихся в мероприятиях (викторинах, соревнованиях), защиты проектов, решения задач поискового характера. Итоги реализации программы могут подводиться в виде итогового контроля следующих форм: защита индивидуального или группового проекта в виде публичного выступления с демонстрацией проектной работы; выставка; соревнование; взаимооценка учащимися работ друг друга. В процессе проведения итогового контроля оценивается результативность освоения программы.

По завершению обучения по данной программе учащиеся получают свидетельство об освоении программы.

### *Характеристика оценочных материалов*

Виды аттестации, сроки проведения	Цель	Содержание	Форма
Входной контроль (сентябрь)	Определить исходный уровень подготовленности учащихся.	Введение в деятельность. Входящая диагностика.	Собеседование. Тестирование.

Текущий контроль успеваемости на каждом занятии (в течение года)	Определить уровень понимания изучаемого материала и уровень приобретенных умений и навыков.	Проверка усвоения материала по теме занятия или комплексу занятий.	Рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий.
Текущий контроль (по итогам изучения разделов)	Определить уровень приобретенных умений и навыков.	Решение практических задач	Тестирование. Практические задания
Итоговая аттестация по итогам 1 года обучения (май)	Определить уровень усвоения достижения планируемых результатов программы в целом	Комплексная творческая работа.	Выполнение задания на основе предлагаемой проблемной задачи.

### Оценочные материалы

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля освоения программы являются внешние образовательные продукты учащихся (созданные сайты, веб-приложения), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам программы.

Оценка достижения планируемых результатов освоения программы осуществляется по 3 уровням:

- высокий (от 81% до 100% освоения программного материала);
- средний (от 51% до 80% освоения программного материала);
- низкий (от 0% до 50% освоения программного материала и менее).

Отслеживание личностного развития учащихся осуществляется методом наблюдения. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся.



## Список литературы

### *Для педагога:*

1. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. Серия Бестселлеры O'Reilly, изд. СПб: Питер, 2014 г. 544 с.
2. Макфарланд Д. Новая большая книга CSS. Серия Бестселлеры O'Reilly, изд. СПб: Питер, 2019 г. 720 с.
3. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. Серия Бестселлеры O'Reilly, изд. СПб: Питер, 2017 г. 768 с.
4. Роббинс Д. HTML5. Карманный справочник. Изд. Вильямс, 2019 г. 192 с.
5. Симдянов И. В. Котеров Д. В., PHP 7 Наиболее полное руководство. Изд. БХВ-Петербург 2017 г. 1088 с.
6. Флэнаган Д. JavaScript. Полное руководство. Изд. Вильямс, 2021 г. 720 с.
7. Фримен Э., Робсон Э. Изучаем программирование на JavaScript. Серия Head First O'Reilly, изд. СПб: Питер, 2016 г. 640 с.

### *Для учащихся:*

1. Кириченко А. В., Хрусталеv А. А. HTML5 + CSS3. Основы современного WEB-дизайна. Изд. Наука и техника, 2018 г. 352 с.
2. МакГрат М., PHP7 для начинающих с пошаговыми инструкциями. Изд. Эксмо-Пресс, 2019 г. 256 с.
3. Морган Н. JavaScript для детей. Самоучитель по программированию. Изд. Манн, Иванов и Фербер. 2016 г. 288 с.

### Календарный учебный график программы

№	Тема занятия	Количество часов			Дата проведения занятия (план)	Дата проведения занятия (факт)
		всего	теория	практика		
1. Основы верстки веб-страниц. HTML и CSS						
1.1	Введение. Техника безопасности. Правила поведения на занятиях. Входной контроль.	1	1	0		
1.2	Знакомство с общими понятиями о веб-разработке. Настройка рабочего окружения.	2	1	1		
1.3	Настройка рабочего окружения. Введение в HTML.	3	1	2		
1.4	Вложенные теги. Атрибуты тегов. Блочные и строчные теги.	3	1	2		
1.5	Работа с изображениями в HTML.	3	1	2		
1.6	Вёрстка списков и таблиц в HTML.	3	1	2		
1.7	Создание веб-форм.	3	1	2		
1.8	Обзор HTML5-тегов.	3	1	2		
1.9	Практика создания полноценной HTML-разметки.	3	0	3		
1.10	Знакомство с каскадной таблицей стилей CSS. Применение простейших свойств стилизации HTML-разметки.	3	1	2		
1.11	Селекторы или способы обращения к элементам страницы для задания стилей. Комбинированные селекторы.	3	1	2		
1.12	Блочная модель CSS.	3	1	2		
1.13	Позиционирование в CSS.	3	1	2		
1.14	Псевдо-классы и псевдо-элементы в CSS.	3	1	2		
1.15	Практика создания веб-страниц со структурой и оформлением.	3	0	3		

1.16	Переходы и анимация в CSS.	3	1	2		
<b>ВСЕГО:</b>		<b>45</b>	<b>14</b>	<b>31</b>		
<b>2. Продвинутая верстка веб-страниц.</b>						
2.1	Основы верстки по макету.	3	1	2		
2.2	Создание структуры веб-страниц.	3	1	2		
2.3	Технология flexbox-верстки.	3	1	2		
2.4	Введение в резиновую и адаптивную верстку. Медиа-запросы в CSS.	3	1	2		
2.5	Верстка резиновых и адаптивных изображений.	3	1	2		
2.6	Верстка резиновых и адаптивных веб-страниц.	3	1	2		
2.7	Знакомство с фреймворками для быстрого построения адаптивных страниц.	3	1	2		
2.8	Понятие сетки для верстки. Применение фреймворков в работе.	3	1	2		
2.9	Практика создания адаптивных страниц с использованием фреймворков.	3	0	3		
<b>ВСЕГО:</b>		<b>27</b>	<b>8</b>	<b>19</b>		
<b>3. Основы JavaScript. Программирование на стороне клиента.</b>						
3.1	Введение в программирование на JavaScript.	3	1	2		
3.2	Переменные. Типы данных. Арифметика.	3	1	2		
3.3	Алгоритмы ветвления.	3	1	2		
3.4	Циклические алгоритмы.	3	1	2		
3.5	Массивы JavaScript.	3	1	2		
3.6	Функции в JavaScript.	3	1	2		
3.7	Объекты в JavaScript.	3	1	2		
<b>ВСЕГО:</b>		<b>21</b>	<b>7</b>	<b>14</b>		
<b>4. JavaScript в браузере.</b>						
4.1	Введение в DOM. Взаимодействие JavaScript-программы с веб-страницей. Получение и изменение элементов.	3	1	2		

4.2	DOM. Навигация по дереву элементов страницы при помощи JavaScript.	3	1	2		
4.3	Изменение структуры HTML-документа при помощи JavaScript.	3	1	2		
4.4	События в JavaScript.	3	1	2		
4.5	Работа с HTML-формами через JavaScript.	3	1	2		
4.6	Асинхронные запросы. Работа с API.	3	1	2		
4.7	Хранение состояния и данных на клиенте.	3	1	2		
4.8	Подключение сторонних модулей и библиотек.	3	2	1		
4.9	Реализация полноценной frontend части веб-сайта.	3	0	3		
<b>ВСЕГО:</b>		<b>27</b>	<b>9</b>	<b>18</b>		
<b>5. Разработка динамических веб-приложений.</b>						
5.1	Введение в тему создания SPA приложений.	3	1	2		
5.2	Работа с Canvas. Создание и управление холстом для игровых веб-приложений.	3	1	2		
5.3	Реализация Drag-and-drop технологии.	3	1	2		
5.4	Изучение и реализация основных игровых механик.	3	1	2		
5.5	Практика создания простейших веб-игр.	3	0	3		
<b>ВСЕГО:</b>		<b>15</b>	<b>4</b>	<b>11</b>		
<b>6. Серверная часть сайтов и веб-приложений (backend PHP+SQL).</b>						
6.1	Введение в разработку backend части веб-проектов.	3	1	2		
6.2	Введение в язык PHP.	3	1	2		
6.3	Условные операторы и алгоритмы ветвления.	3	1	2		
6.4	Циклические алгоритмы.	3	1	2		
6.5	Массивы.	3	1	2		

6.6	Ассоциативные и многомерные массивы.	3	1	2		
6.7	Функции.	3	1	2		
6.8	Взаимодействие PHP с веб-страницей. Работа с веб-формами.	3	1	2		
6.9	Работа с sessions и cookies. Хранение данных на стороне сервера.	3	1	2		
6.10	Введение в базы данных. Язык запросов SQL.	3	1	2		
6.11	Создание таблиц. Основные запросы к базе данных.	3	1	2		
6.12	Проектирование связанных таблиц.	3	1	2		
6.13	Введение в объектно-ориентированное программирование в PHP.	3	1	2		
6.14	Наследование и полиморфизм.	3	1	2		
6.15	Проектирование классов.	3	1	2		
6.16	Практика создания full stack веб-приложения.	3	0	3		
<b>ВСЕГО:</b>		<b>48</b>	<b>15</b>	<b>33</b>		
<b>7. Проектная деятельность</b>		<b>21</b>	<b>3</b>	<b>18</b>		
<b>8. Повторение изученного материала, обобщение, подготовка к итоговому контролю</b>		<b>3</b>	<b>3</b>	<b>-</b>		
<b>9. Презентация и защита проекта. Итоговый контроль</b>		<b>9</b>	<b>3</b>	<b>6</b>		
<b>ИТОГО:</b>		<b>216</b>	<b>66</b>	<b>150</b>		

## Лабораторные работы

**Название:** Лабораторная работа: “Создание анимации и переходов в CSS”

**Автор:** Бушенев Игорь Владимирович

**Направленность:** Техническая

**Цель:** создать собственные анимации и переходы при помощи CSS

**Задачи:**

- Изучение основ CSS
- Создание анимации на CSS
- Разработка структуры анимации
- Тестирование анимации
- Исправление ошибок

**Артефакт:** Работающая анимация на CSS

УСЛОВИЯ РЕАЛИЗАЦИИ		
Время	Форма	Методы
3 ак. ч.	Групповая и индивидуальная	Практический, интерактивный

КАТЕГОРИЯ ОБУЧАЮЩИХСЯ		
Возраст	Особенности	Количество участников в группе
14-17	Учащиеся, интересующиеся программированием сайтов и приложений	12

РЕСУРСЫ		
Оборудование и материалы	Базовые навыки	Уровень сложности и место в структуре курса
Компьютеры или ноутбуки с доступом в Интернет, текстовые	Знание основ HTML и CSS	Средняя. Тема занимает место крайнего занятия в разделе «основы верстки»

редакторы		
-----------	--	--

<b>ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ</b>		
Hard-skills, предметные навыки	Soft-Skills, метапредметные навыки	Личностные
умение создавать анимацию в CSS, знание HTML и CSS	креативность, внимание к деталям, навыки анализа.	Ответственность, самостоятельность, умение работать с информацией

<b>ОПЫТ РЕАЛИЗАЦИИ</b>		
Инструкции для педагога	Открытая информация о результатах	Участие в конкурсах
<p>Познакомьте учащихся с темой “Создание и анимация CSS” и объясните ее важность и актуальность.</p> <p>Раздайте учащимся практические работы. Помогите учащимся изучить основы CSS и создайте условия для создания и анимации CSS на практике.</p> <p>Используйте примеры и практические задания для минимизации возможных рисков.</p> <p>Подведите итоги занятия, обсудите результаты и ответьте на вопросы учащихся.</p>	Результаты данной лабораторной работы будут доступны учащимся после ее завершения	Данная лабораторная работа не предполагает участие в конкурсах.

**ЭТАП 1 Постановка цели:**

Цель данного этапа - познакомить участников с темой “Создание анимации CSS”. Рекомендованное время - 10 минут. Задание/Активность - Обсуждение темы, объяснение ее важности и актуальности. Рисков на этом этапе нет.

### **ЭТАП 2 Основная часть:**

Цель - обучение участников созданию анимации CSS. Рекомендованное время - 1 час 30 минут.

Задания/Активность - изучение основ CSS, создание анимации, разработка структуры анимации, тестирование анимации, исправление ошибок.

Возможные риски - сложность материала, отсутствие мотивации. Варианты минимизации рисков - разбиение материала на небольшие части, активное использование примеров и практических заданий.

### **ЭТАП 3 Рефлексия/итоги:**

Цель - подведение итогов и закрепление полученных знаний.

Рекомендованное время - 20 минут.

Задач/Активность - обсуждение результатов, ответы на вопросы, выдача домашнего задания.

Возможные риски - недостаток времени, усталость участников. Варианты минимизации рисков - ограничение времени на обсуждение, подготовка заранее к вопросам участников.

## **ИСПОЛЬЗОВАННЫЕ РЕСУРСЫ**

Для проведения работы были использованы следующие ресурсы:

- 1) Учебные материалы по CSS
- 2) Интернет-ресурсы для поиска информации по теме
- 3) Текстовый редактор для создания кода

## **ДОПОЛНИТЕЛЬНЫЕ РЕСУРСЫ**

Для педагогов:

Справочник по CSS: <https://developer.mozilla.org/ru/docs/Web/CSS/Reference>

Примеры кода CSS на CodePen: <https://codepen.io/>

Для учащихся:

Учебник по основам CSS: <https://learn.javascript.ru/css-basics>

Интерактивный учебник по CSS: <http://htmlbook.ru/blog/kurs-po-css>



**Название:** Создание веб-форм на основе HTML

**Автор:** Бушенев Игорь Владимирович

**Направленность:** Техническая

**Цель:** Обучение учащихся созданию веб-форм с применением HTML

**Задачи:**

- 1) Изучение основ HTML
- 2) Создание простой веб-формы
- 3) Разработка структуры веб-формы
- 4) Тестирование веб-формы на наличие ошибок
- 5) Исправление возможных ошибок

**Артефакт:** Рабочая веб-форма, созданная на основе HTML

УСЛОВИЯ РЕАЛИЗАЦИИ		
Время	Форма	Методы
3 ак. ч.	Индивидуальная	Практический, интерактивный

КАТЕГОРИЯ ОБУЧАЮЩИХСЯ		
Возраст	Особенности	Количество участников в группе
14-17	Учащиеся, интересующиеся программированием сайтов и приложений	12

РЕСУРСЫ		
Оборудование и материалы	Базовые навыки	Уровень сложности и место в структуре курса
Оборудование и материалы: Компьютер или ноутбук с доступом в интернет, текстовый редактор (Notepad++, Sublime Text и т.д.)	Знание основ HTML и CSS	Начальный. Тема занимает место срединного занятия в разделе «основы верстки»

<b>ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ</b>		
Hard-skills, предметные навыки	Soft-Skills, метапредметные навыки	Личностные
Умение создавать веб-формы с использованием HTML, знание основ HTML	Креативность, внимание к деталям, аналитические навыки	Ответственность, самостоятельность, умение работать с информацией

<b>ОПЫТ РЕАЛИЗАЦИИ</b>		
Инструкции для педагога	Открытая информация о результатах	Участие в конкурсах
Познакомьте учеников с целью лабораторной работы, объясните важность создания веб-форм. Разделите учеников на группы или проведите урок в форме семинара. Помогите ученикам изучить основы HTML и создать веб-форму, проведите тестирование и исправление ошибок. Подведите итоги занятия и обсудите результаты с учениками.	Результаты данной работы могут быть представлены учениками в качестве портфолио или демонстрационного материала	Лабораторная работа не предусматривает участие в конкурсах, однако ее результаты могут быть представлены на различных конкурсах и соревнованиях

## **ЭТАПЫ:**

### **Этап 1. Постановка цели:**

Рекомендованное время: 10 мин.

Задание/Активность: Обсуждение темы, целей и задач работы.

Риски этапа: Отсутствие мотивации у учеников.

Варианты минимизации рисков: Создание интересной и актуальной темы для обсуждения.

## **Этап 2. Основная часть:**

Рекомендованное время: 95 мин.

Задание/Активность: Изучение основ HTML, создание веб-формы, тестирование и исправление ошибок.

Риски этапа: Сложность материала, технические проблемы.

Варианты минимизации рисков: Разделение материала на небольшие блоки, использование примеров и практического подхода.

## **Этап 3. Рефлексия/итоги:**

Рекомендованное время: 15 мин.

Задание/Активность: Подведение итогов, обсуждение результатов.

Риски этапа: Нехватка времени, утомление учеников.

Варианты минимизации рисков: Ограничение времени на обсуждение, заблаговременная подготовка к вопросам учеников.

## **ИСПОЛЬЗОВАННЫЕ РЕСУРСЫ**

Для проведения лабораторной работы используются учебные материалы по HTML, интернет-ресурсы для поиска информации, текстовый редактор для создания веб-форм.

## **ДОПОЛНИТЕЛЬНЫЕ РЕСУРСЫ**

Для педагога:

1. Справочник по HTML: <https://htmlacademy.ru/>
2. Примеры кода HTML: <https://www.w3schools.com/html/default.asp>

Для учеников:

1. Учебник по основам HTML: <https://webref.ru/html>
2. Интерактивный учебник по HTML: <http://htmllivebook.ru/>.

**Название:** Блочная модель CSS

**Автор:** Бушенев Игорь Владимирович

**Направленность:** Техническая

**Цель:** обучить учащихся использованию блочной модели CSS для создания структурированных веб-страниц

**Задачи:**

- 1) Изучить основные принципы блочной модели CSS
- 2) Освоить свойства CSS для управления блочной моделью
- 3) Разработать структуру веб-страницы на основе блочной модели
- 4) Протестировать код на наличие ошибок

**Артефакт:** Структурированная веб-страница, созданная с использованием блочной модели CSS

УСЛОВИЯ РЕАЛИЗАЦИИ		
Время	Форма	Методы
3 ак. ч.	Групповая или индивидуальная	Практические, интерактивные, исследовательские

КАТЕГОРИЯ ОБУЧАЮЩИХСЯ		
Возраст	Особенности	Количество участников в группе
14-17	Учащиеся, интересующиеся программированием сайтов и приложений	12

РЕСУРСЫ		
Оборудование и материалы	Базовые навыки	Уровень сложности и место в структуре курса
Интернет, компьютер, текстовый редактор кода	Основные принципы CSS	Высокий. Тема является основополагающей в разделе «основы верстки»

<b>ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ</b>		
Hard-skills, предметные навыки	Soft-Skills, метапредметные навыки	Личностные
Знать основные понятия и принципы CSS. Уметь применять свойства блочной модели. Владеть навыками структурирования веб-страницы. опыт тестирования кода на наличие ошибок.	Умение работать в команде. Навыки критического мышления. анализировать и решать проблемы.	Развитие интереса к программированию и веб-дизайну. навыков самостоятельного обучения и саморазвития.

<b>ОПЫТ РЕАЛИЗАЦИИ</b>		
Инструкции для педагога	Открытая информация о результатах	Участие в конкурсах
Перед началом работы необходимо объяснить учащимся цели и задачи работы, а также обсудить основные принципы и понятия блочной модели CSS. Рекомендуется использовать демонстрационные примеры и практический подход для лучшего понимания материала.	Результаты работы могут быть представлены учащимися в виде структурированной веб-страницы или портфолио.	Лабораторная работа не предусматривает участие в конкурсах

## **ЭТАПЫ:**

### **Этап 1. Постановка цели:**

Рекомендованное время: 10 мин

Задание/Активность: Обсуждение целей и задач работы, основных понятий и принципов CSS.

Риски этапа: Отсутствие мотивации у учащихся.

Варианты минимизации рисков: Использование демонстрационных примеров и практического подхода для лучшего понимания материала.

### **Этап 2. Основная часть:**

Рекомендованное время: 90 мин

Задание/Активность: Применение свойств блочной модели, структурирование веб-страницы и тестирование кода на наличие ошибок.

Риски этапа: Сложность материала может вызвать затруднения у учащихся.

Варианты минимизации рисков: Разделение материала на более мелкие и понятные части, предоставление примеров и обратной связи.

### **Этап 3. Рефлексия/итоги:**

Рекомендованное время: 20 мин

Задание/Активность: Анализ проделанной работы, обсуждение результатов и возможных улучшений.

Риски этапа: Ученики могут испытывать усталость и недостаток мотивации после длительной работы.

Варианты минимизации рисков: Заранее подготовить вопросы для обсуждения, ограничить время на обсуждение.

## **ИСПОЛЬЗОВАННЫЕ РЕСУРСЫ**

- Учебные материалы по основам CSS.
- Интернет-ресурсы для получения дополнительной информации и примеров кода.
- Текстовый редактор для создания и редактирования кода CSS.

## **ДОПОЛНИТЕЛЬНЫЕ РЕСУРСЫ**

Для педагога:

Справочник по CSS: <https://developer.mozilla.org/ru/docs/Web/CSS>

Примеры кода CSS: <https://www.w3schools.com/css/default.asp>

Онлайн-редактор кода CSS: <http://jsfiddle.net/>

Для обучающихся:

Учебник по CSS: <https://learn.javascript.ru/css-basics>

Онлайн-обучение CSS: <https://stepik.org/course/34/>

Интерактивные уроки по CSS: <http://lessons.school/>

**Название:** Верстка списков и таблиц в HTML

**Автор:** Бушенев Игорь Владимирович

**Направленность:** Техническая

**Цель:** научить учащихся создавать и верстать списки и таблицы с использованием HTML.

**Задачи:**

1. Познакомить учащихся с основными тегами HTML для создания списков и таблиц.
2. Развить навыки работы с кодом HTML.
3. Сформировать понимание принципов и методов верстки веб-страниц.

**Артефакт:** Веб-страница с корректно отображаемыми списками и таблицами, созданная с помощью HTML и CSS.

УСЛОВИЯ РЕАЛИЗАЦИИ		
Время	Форма	Методы
3 ак. ч.	Групповая или индивидуальная	Практические, интерактивные, исследовательские

КАТЕГОРИЯ ОБУЧАЮЩИХСЯ		
Возраст	Особенности	Количество участников в группе
14-17	Учащиеся, интересующиеся программированием сайтов и приложений	12

РЕСУРСЫ		
Оборудование и материалы	Базовые навыки	Уровень сложности и место в структуре курса
Интернет, компьютер, текстовый редактор кода	Основные принципы HTML	Средний. Тема является основополагающей в разделе «основы верстки». Является некоторым завершающим этапом в изучении языка HTML

<b>ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ</b>		
Hard-skills, предметные навыки	Soft-Skills, метапредметные навыки	Личностные
Умение создавать и верстать веб-страницы с использованием списков и таблиц, знание синтаксиса HTML	Работа в команде, коммуникация, критическое мышление	Развитие интереса к веб-разработке, формирование навыков самообучения и саморазвития

<b>ОПЫТ РЕАЛИЗАЦИИ</b>		
Инструкции для педагога	Открытая информация о результатах	Участие в конкурсах
<p>Перед началом работы следует объяснить учащимся цель и задачи работы, обсудить основные теги и атрибуты HTML, используемые для создания списков и таблиц. Дать практические задания на вёрстку таблиц разного уровня сложности.</p> <p>Продемонстрировать и закрепить на практике многоуровневые списки и сложно структурированные таблицы.</p> <p>Предоставьте учащимся время для самостоятельной работы, а затем проведите обсуждение результатов</p>	Результаты работы могут быть предоставлены учащимися в виде созданной веб-страницы	Лабораторная работа не предусматривает участие в конкурсах



## **ЭТАПЫ:**

### **Этап 1. Постановка цели:**

Рекомендованное время: 10 минут

Задание/Активность: Определение цели и задач урока, обсуждение ожидаемых результатов

Риски этапа: Потеря интереса учащихся к теме

Варианты минимизации рисков: Введение в тему с использованием интересных примеров и задач

### **Этап 2. Основная часть:**

Рекомендованное время: 95 минут

Задание/Активность: Изучение и применение на практике тегов и атрибутов HTML для создания и стилизации списков и таблиц

Риски этапа: Сложности в понимании материала и его применении

Варианты минимизации рисков: Предоставление подробных инструкций и примеров, разделение материала на небольшие блоки

### **Этап 3. Рефлексия/итоги:**

Рекомендованное время: 15 минут

Задание/Активность: Обсуждение результатов работы, анализ ошибок и успехов

Риски этапа: Учащиеся могут испытывать усталость

Варианты минимизации рисков: Ограничение времени на обсуждение, заранее подготовленные вопросы

## **ИСПОЛЬЗОВАННЫЕ РЕСУРСЫ**

Учебник по HTML и CSS: <https://htmlacademy.ru/>

Примеры кода HTML и CSS: <https://codepen.io/>

## **ДОПОЛНИТЕЛЬНЫЕ РЕСУРСЫ**

Для педагогов:

Справочник по HTML: <https://webref.ru/html>

Примеры кода HTML: <https://htmldocs.ru/>

Онлайн-редактор кода HTML: <http://htmledit.squarefree.com/>

Для учащихся:

Учебник по HTML: <https://ru.hexlet.io/courses/profession-web-developer/tracks/html>

Интерактивные уроки по HTML: <http://www.codecademy.com/learn/learn-html>

**Название:** Лабораторная работа “Псевдоклассы и псевдоэлементы в CSS”

**Автор:** Бушенев Игорь Владимирович

**Направленность:** Техническая

**Цель:** Изучение и практическое применение псевдоклассов и псевдоэлементов в CSS для создания стилизованных веб-страниц.

**Задачи:**

1. Изучение основных псевдоклассов и их свойств.
2. Изучение основных псевдоэлементов и их свойств.
3. Применение псевдоклассов и псевдоэлементов для стилизации веб-страницы.
4. Создание и оформление веб-страницы, используя псевдоклассы и псевдоэлементы.

**Артефакт:** Созданная веб-страница с использованием псевдоклассов и псевдоэлементов.

УСЛОВИЯ РЕАЛИЗАЦИИ		
Время	Форма	Методы
3 ак. ч.	Групповая или индивидуальная	Практические, интерактивные

КАТЕГОРИЯ ОБУЧАЮЩИХСЯ		
Возраст	Особенности	Количество участников в группе
14-17	Учащиеся, интересующиеся программированием сайтов и приложений	12

РЕСУРСЫ		
Оборудование и материалы	Базовые навыки	Уровень сложности и место в структуре курса
Интернет, компьютер, текстовый редактор кода	Знания HTML и CSS на базовом уровне.	Высокий. Тема является одной из финальных в разделе «основы верстки».

<b>ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ</b>		
Hard-skills, предметные навыки	Soft-Skills, метапредметные навыки	Личностные
Знание основных псевдоклассов и псевдоэлементов CSS, умение применять их для стилизации вебстраниц	Умение работать в команде, коммуникация, критическое мышление	Развитие интереса к веб-разработке, формирование навыков самообразования и саморазвития, развитие креативности и творческого мышления.

<b>ОПЫТ РЕАЛИЗАЦИИ</b>		
Инструкции для педагога	Открытая информация о результатах	Участие в конкурсах
Разъяснить учащимся цели и задачи работы, объяснить, что такое псевдоклассы и псевдоэлементы, как их использовать, распределить роли в группе и предоставить время на самостоятельную работу.	Результаты работы могут быть предоставлены учащимися в виде созданной веб-страницы	Лабораторная работа не предусматривает участие в конкурсах

## **ЭТАПЫ:**

### **Этап 1. Постановка цели:**

Рекомендованное время - 10 мин

Задание/активность - Определение целей и задач работы, обсуждение ожидаемых результатов

Риски - Потеря интереса учащихся

Варианты минимизации - Введение в тему с примерами и задачами

### **Этап 2. Основная часть:**

Рекомендованное время - 95 мин

Задание/активность - Изучение и использование псевдоклассов и псевдоэлементов, создание и стилизация веб-страниц

Риски - Сложность материала, непонимание

Варианты минимизации - Подробное объяснение материала, разбиение на блоки

### **Этап 3. Рефлексия/итоги:**

Рекомендованное время – 15 мин

Задание/активность - Обсуждение результатов, анализ успехов и ошибок

Риски - Усталость учащихся

Варианты минимизации - Ограничение времени обсуждения, подготовка вопросов

## **ИСПОЛЬЗОВАННЫЕ РЕСУРСЫ**

Учебник по HTML и CSS: <https://htmlacademy.ru/>

Примеры кода HTML и CSS: <https://codepen.io/>

## **ДОПОЛНИТЕЛЬНЫЕ РЕСУРСЫ**

Для педагога:

Справочник по CSS: <https://developer.mozilla.org/ru/docs/Web/CSS>

Примеры кода CSS: <https://www.w3schools.com/css/default.asp>

Онлайн-редактор кода CSS: <http://jsfiddle.net/>

Для обучающихся:

Учебник по CSS: <https://learn.javascript.ru/css-basics>

Онлайн-обучение CSS: <https://stepik.org/course/34/>

Интерактивные уроки по CSS: <http://lessons.school/>