

Дополнительная общеобразовательная
общеразвивающая программа

«Веб-разработка»

Возраст учащихся: 12-16 лет

Срок реализации: 1 год

Разработчик:
Курышев Валентин Владимирович

Пояснительная записка

Программа дополнительного образования «Веб-разработка» имеет техническую направленность.

Программа разработана в соответствии с современными нормативными документами в сфере образования:

- *Федерального государственного образовательного стандарта основного общего образования, утвержденного приказом Министерства образования и науки Российской Федерации от 17.12.2010 № 1897 (далее - ФГОС основного общего образования);*

- Порядка организации и осуществления образовательной деятельности по основным общеобразовательным программам - образовательным программам начального общего, основного общего и среднего общего образования, утвержденного приказом Минпросвещения России от 28.08.2020 № 442;

- Федерального перечня учебников, допущенных к использованию при реализации имеющих государственную аккредитацию образовательных программ начального общего, основного общего, среднего общего образования организациями, осуществляющими образовательную деятельность, утвержденного приказом Минпросвещения России от 20.05.2020 №254;

- Перечня организаций, осуществляющих выпуск учебных пособий, которые допускаются к использованию при реализации имеющих государственную аккредитацию образовательных программ начального общего, основного общего, среднего общего образования, утвержденного приказом Министерства образования и науки Российской Федерации от 09.06.2016 № 699;

- Санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденных постановлением Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 (далее - СП 2.4.3648-20);

- Санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания», утвержденных постановлением Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 (далее - СанПиН 1.2.3685-21);

- Распоряжения Комитета по образованию от 12.04.2021 № 1013-р «О формировании календарного учебного графика государственных образовательных учреждений Санкт-Петербурга, реализующих основные общеобразовательные программы, в 2021/2022 учебном году»;

- Распоряжения Комитета по образованию от 09.04.2021 № 997-р «О формировании учебных планов государственных образовательных учреждений Санкт-Петербурга, реализующих основные общеобразовательные программы, на 2021/2022 учебный год»;

- Основной образовательной программой среднего общего образования ГБОУ школа №525 с углубленным изучением английского языка дважды Героя Советского Союза Г.М. Гречко Московского района Санкт-Петербурга, утвержденной приказом № 47 от 21 мая 2020г. ;

- Уставом ГБОУ школа №525 с углубленным изучением английского языка дважды Героя Советского Союза Г.М. Гречко Московского района Санкт-Петербурга;

Актуальность программы заключается в том, что она определяется запросом общеобразовательных организаций в части расширения и углубления знаний в предметной области информатика, формированием инженерно-технологического мышления, формированием прикладных компетенций цифровой экономики. Педагогическая целесообразность программы - определяется необходимостью подготовки учащихся к дальнейшему обучению в профильных классах, к участию в предметных олимпиадах различного уровня, дает возможность применить свои знания в прикладных областях. Программа рассчитана на школьников, желающих освоить работу с языками программирования высокого уровня. Программа способствует созданию условий для социального, культурного и профессионального самоопределения, творческой самореализации личности ребенка, его будущей интеграции в системе мировой и российской науки; популяризации профессий, связанных с программированием. Программа дополнительного образования «Веб-разработка» позволит познакомить учащихся с основами программирования и показать весь спектр применения программирования в современном мире.

Отличительные особенности программы : имеет техническую направленность и нацелена, в первую очередь, на изучение IT дисциплин и формирование практических навыков программирования и решения задач по программированию с использованием процедурного и объектно-ориентированного программирования на языке JavaScript.

Адресат программы: Возраст учащихся 13 – 16 лет. На обучение по программе принимаются все желающие, имеющие элементарные навыки работы на компьютере и не имеющие противопоказаний по состоянию здоровья для работы на ПК.

Объем и срок реализации программы: 144 часа на 1 год обучения.

Цель программы – углубление и расширение знаний и навыков учащихся в предметной области информатика, формирование компетенций в области разработки алгоритмов и программ с использованием процедурного и объектно-ориентированного программирования в средах веб-разработки, формирование сквозных компетенций по цифровому образованию.

Задачами программы являются:

Образовательные:

- сформировать устойчивые базовые навыки программирования, алгоритмизации языка программирования JavaScript, включая языки разметки и таблиц стилей (HTML и CSS);
- освоить основные алгоритмы обработки данных, способов решения задач;
- сформировать навыки работы с дополнительными библиотеками и расширениями языка программирования JavaScript (React.js, Node.js, TypeScript и т. д.) в составе сред разработки (пользовательские интерфейсы, работа с данными);
- расширение знаний для решения задач по программированию и алгоритмизации;
- формирование представлений о виртуальной, дополненной и смешанной реальности, базовых понятиях, актуальности и перспективах данных технологий;
- формирование навыков проектной работы;
- формирование информационной культуры как совокупности знаний, умений, информационного мировоззрения и информационного поведения, необходимых в современном обществе.

Развивающие:

- развитие творческого и инженерного мышления при постановке прикладных задач и их подведение к реализации на ПК и в сети интернет;
- развитие логического и алгоритмического мышления;
- развитие интереса к изучению информатики и программирования;
- развитие межпредметных умений и навыков;
- развитие сквозных компетенций по цифровой экономике;
- развитие умений управлять проектами и процессами;
- развитие коммуникативных компетенций: навыков работы с различными источниками информации, навыками самостоятельного поиска, отбора и применения информации, необходимой для решения учебных задач;

Воспитательные:

- повышение мотивации к проектированию;
- формирование стремления к получению качественного законченного результата;
- формирование навыков проектного мышления и работы в команде - воспитывать лидерские качества, ответственность, целеустремленность и организованность.

Условия реализации программы: данная программа рассчитана на 1 год обучения. Набор детей в группы свободный, ограниченный лишь медицинскими показателями. Состав группы постоянный.

Объем программы – 144 часа. Сроки реализации программы: 1 год. Режим занятий: 2 раза в неделю по 2 часа.

Наполняемость групп: 10 человек.

Реализация программы предусматривает включение **здоровье сберегающего компонента:** упражнений для глаз, физкультминуток, чередование практической работы на

компьютере с теоретическими занятиями, беседами, работой в тетради, обсуждениями, экскурсиями.

Кадровое обеспечение: педагоги, ведущие занятия должны иметь опыт ведения кружка, связанным с программированием.

Эффективная реализация программы требует технического оборудования: компьютеры или ноутбуки, операционная система Windows, доступ в интернет для каждого ученика.

Планируемые результаты обучения

Личностные результаты:

- сформированность познавательных интересов, интеллектуальных и творческих способностей учащихся;
- мотивация образовательной деятельности школьников на основе личностно-ориентированного подхода;
- формирование ценностных отношений друг к другу, учителю, авторам открытий и изобретений, результатам обучения.

Метапредметные результаты:

- овладение навыками самостоятельного приобретения новых знаний;
- овладение навыками организации учебной деятельности, постановки целей, планирования, самоконтроля и оценки результатов своей деятельности, умениями предвидеть возможные результаты своих действий;
- формирование умений воспринимать, перерабатывать и предъявлять информацию в словесной, образной, символической формах;
- формирование умений анализировать и перерабатывать полученную информацию в соответствии с поставленными задачами, выделять основное содержание прочитанного текста, находить в нем ответы на поставленные вопросы и излагать его;
- формирование ИКТ-компетентности.

Предметные результаты:

Правила поведения в компьютерном классе.

Учащийся научится:

- соблюдать требования к организации внутреннего пространства компьютера и рабочего места;
- выполнять правила поведения в компьютерном классе.

Учащийся получит возможность научиться:

- соблюдать требования безопасности и гигиены в работе со средствами ИКТ.

Аппаратные средства современных персональных компьютеров (ПК).

Учащийся научится:

- работать с кодом;
- понимать «чужой» код.

Учащийся получит возможность научиться:

- разрабатывать собственные сайты и приложения.

Программные средства для разработки.

Учащийся научится:

- работать в среде Visual Studio;
- новым технологиям в HTML и CSS;
- разрабатывать приложения на языке программирования JavaScript.

Учащийся получит возможность научиться:

- создавать приложения в любой среде разработки и браузере.

Календарно-тематическое планирование

№ п/п	Название раздела, темы	Количество часов			Формы контроля
		Всего	Теория	Практика	
1.	Введение в веб-разработку. Основы HTML и CSS. Строчные и блочные элементы	4 ч	2 ч	2 ч	Практическая работа. Опрос.
2.	Позиционирование Flexbox Grid Layout. Препроцессоры CSS. Практика создания адаптивного дизайна	4 ч	1 ч	3 ч	Практическая работа. Анализ работ.
3.	Методология БЭМ и анимация в приложениях	4 ч	-	4 ч	Практическая работа
4.	Знакомство с JavaScript. Условное ветвление и основы функций. Циклы и массивы	4 ч	2 ч	2 ч	Тестирование. Практическая работа
5.	Продвинутая работа с циклами и массивами. Работа с объектами. Подготовка к индивидуальному проекту	4 ч	2 ч	2 ч	Тестирование. Практическая работа
6.	Browser Object Model. Введение в Document Object Model	4 ч	-	4 ч	Тестирование
7.	Шаблонизация	4 ч	2 ч	2 ч	Анализ работ
8.	События	4 ч	-	4 ч	Практическая работа
9.	Работа с формами и элементами на странице	4 ч	2 ч	2 ч	Анализ работ
10.	Типовые виджеты	4 ч	2 ч	2 ч	Тестирование
11.	Аjax и работа с данными	4 ч	2 ч	2 ч	Анализ работ
12.	Работа с API	4 ч	2 ч	2 ч	Тестирование. Практическая работа.
13.	Использование сторонних библиотек и организация кода	4 ч	2 ч	2 ч	Анализ работ
14.	Ретроспектива JSB	4 ч	2 ч	2 ч	Тестирование
15.	Q&A сессия	4 ч	4 ч	-	Анализ работ
16.	Жизненный цикл проекта и основы GIT	4 ч	-	4 ч	Практическая работа. Обсуждение.
17.	Code Review	4 ч	2 ч	2 ч	Тестирование
18.	Пакетные менеджеры. Code Linting	4 ч	4 ч	-	Анализ работ
19.	Инструменты для сборки проектов	4 ч	2 ч	2 ч	Тестирование
20.	Typescript	4 ч	2 ч	2 ч	Анализ работ
21.	Основы тестирования (автоматические тесты, Unit-тесты). Отладка, дебаггинг и профилирование приложения	4 ч	2 ч	2 ч	Тестирование. Практическая работа.
22.	Введение в React. JSX. Жизненный циклы компонент (lifecycle)	4 ч	2 ч	2 ч	Анализ работ
23.	Функциональный компонент. Hooks	4 ч	2 ч	2 ч	Тестирование
24.	PropTypes. Коллекции, обработка событий	4 ч	-	4 ч	Практическая работа. Обсуждение.

25.	Styled components. Контекст (Context API). Redux	4 ч	2 ч	2 ч	Тестирование
26.	Тестирование. Фреймворк Jest	4 ч	-	4 ч	Анализ работ. Практическая работа.
27.	Основы работы бэкенда	4 ч	2 ч	2 ч	Тестирование
28.	Express.js и базы данных	4 ч	2 ч	2 ч	Анализ работ
29.	Rest API	4 ч	2 ч	2 ч	Тестирование
30.	Node.js	4 ч	-	4 ч	Анализ работ. Практическая работа.
31.	Асинхронность. Оптимизация	4 ч	2 ч	2 ч	Тестирование
32.	Формы React Refs	4 ч	2 ч	2 ч	Анализ работ
33.	Q&A сессия	4 ч	2 ч	2 ч	Тестирование
34.	Творческая итоговая работа	4 ч	-	4 ч	Тестирование. Анализ работ.
35.	Подготовка к защите проектов	4 ч	-	4 ч	Тестирование. Анализ работ.
36.	Защита проектов	4 ч	-	4 ч	Наблюдение
	Итого:	144 ч	55 ч	89 ч	

Учебный график

Год обучения	Дата начала обучения по программе	Дата окончания обучения по программе	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год			36	72	144	2 раза в неделю по 2 часа

РАБОЧАЯ ПРОГРАММА

к дополнительной общеобразовательной
общеразвивающей программе
«Веб-разработка»

Год обучения: 1-й

Педагог дополнительного образования:

Курышев Валентин Владимирович

Содержание программы

Тема 1. Введение в веб-разработку. Основы HTML и CSS. Строчные и блочные элементы.

Теория Правила поведения в компьютерном классе. Теория по HTML и CSS.

Практика Установка на ПК Visual Studio. Добавляем изображения и формы в HTML-страницу. Синтаксис HTML. Список горячих клавиш, которые вам пригодятся.

Тема 2. Позиционирование Flexbox Grid Layout. Препроцессоры CSS. Практика создания адаптивного дизайна.

Теория Продвинутое свойства стилей. Figma – правила использования. Строчно-блочные элементы. Позиционирование.

Практика Горизонтальное и вертикальное позиционирование. Центрирование проекта. Примеры позиционирования. Сравнение Flexbox и Grid Layout.

Тема 3. Методология БЭМ и анимация в приложениях

Теория Методология БЭМ (*Как давать логичные имена классам*). Проблемы анимации и их решения.

Практика CSS-переходы (*transition*). Трансформация (*transform*). CSS3-анимация.

Тема 4. Знакомство с JavaScript. Условное ветвление и основы функций. Циклы и массивы

Теория Знакомство с базовыми операторами. Консоль в Google Chrome. Условное ветвление. Встроенные объекты Math().

Практика Подготовка к созданию JS-кода. Переменные. Методы alert, prompt, confirm. Типы данных в JS. Подготовка рабочей среды (*установка плагинов*). Логические операторы. Реализация циклов while и for.

Тема 5. Продвинутое работа с циклами и массивами. Работа с объектами. Подготовка к индивидуальному проекту

Теория Рассмотреть, какие существуют методы работы с массивами, как происходит копирование массива, а также область их применения.

Практика Методы массива. Очередь и стек. Многомерные массивы. Рекурсия.

Тема 6. Browser Object Model. Введение в Document Object Model

Теория Детали работы с DOM - как перемещаться по дереву относительно любого произвольного элемента, использование наиболее удобных и продвинутых способов, выбор элементов с помощью знакомых CSS-селекторов.

Практика Миграция с id на классы. Перемещение по DOM относительно объекта-узла. Коллекции (*HtmlCollection, NodeList, Array*). Обработчик кликов для нескольких элементов. GetComputedStyle(element).

Тема 7. Шаблонизация

Теория Почему лучше не использовать innerHTML. Методы создания верстки в браузере. Добавление элемента в документ.

Практика Создание объекта-узла. Некоторые приемы при добавлении элементов (*в начало и конец списка*). Удаление элемента.

Тема 8. События

Теория Что такое событие и как оно возникает. Как браузер понимает, по какому элементу кликнули. Некоторые популярные события, связанные с мышью, клавиатурой или самим документом. Что такое обработка события браузером по умолчанию. Что такое всплытие и как оно связано с делегированием.

Практика Способы обработки событий. Основные события мыши. Основные события клавиатуры. Основные события элементов форм. Основные события документа и его элементов. Объект события. Всплытие. Делегирование.

Тема 9. Работа с формами и элементами на странице

Теория Как форма отправляется по умолчанию. Какие бывают параметры у полей ввода форм и на что они влияют.

Практика Отправка формы по умолчанию. Action. Передача данных. Типы полей ввода. Основные события для форм (*submit, change, input*).

Тема 10. Типовые виджеты

Теория Как разрабатывать комплексные интерфейсные элементы.

Практика Позиционирование элемента. Перемещение элемента к другому элементу. `ElementFromPoint`.

Тема 11. Ажак и работа с данными

Теория Что браузер может запрашивать при переходе по ссылке. В чем разница между асинхронным и синхронным запросами. Почему всегда следует использовать асинхронные запросы.

Практика Запрос данных браузером. `Asynchronous JavaScript And XML`. `XMLHttpRequest`. Обработка ошибок. Живые виджеты. API.

Тема 12. Работа с API

Теория Что такое API. Как передаются данные в API. Различные способы кодирования данных.

Практика Передача данных в API (Передача данных через AJAX, с помощью URL, в теле запроса). JSON. CORS. Программы для отправки запросов по HTTP.

Тема 13. Использование сторонних библиотек и организация кода

Теория Как раньше искали дополнительные библиотеки и виджеты для своего проекта. Какие бывают менеджеры пакетов для Frontend-проектов.

Практика NPM. Поиск пакета необходимого для проекта. Внедрение найденного пакета.

Тема 14. Ретроспектива JSB

Теория Структурирование пройденного материала.

Практика Разбор выполненных домашних заданий.

Тема 15. Q&A сессия

Теория Инструменты разработки, не вошедшие в данный курс.

Практика Ответы на вопросы учащихся. Оптимизация кода.

Тема 16. Жизненный цикл проекта и основы GIT

Теория Жизненный цикл Frontend-проектов. Состав продуктовой команды. Что такое Git.

Практика SSH-ключи. Commit. Создание репозитория локально. Merge request и Pull Request. Обновление локальной кодовой базы.

Тема 17. Code Review

Теория Особенности Code Review.

Практика Создание и принятие Pull Request.

Тема 18. Пакетные менеджеры. Code Linting

Теория Npm (Nodejs Package Manager). Основные команды для работы. Устройство package.json. Линтеры и формтеры кода.

Практика Packages. Пакетный менеджер npm. Инициализация фронтového приложения. Стандарты в программировании. Формтеры. Стандарты в программировании. Линтеры.

Тема 19. Инструменты для сборки проектов

Теория Webpack.

Практика Подробнее про работу и задачи webpack. Запуск webpack. Плагины (plugins). Оптимизация.

Тема 20. Typescript

Теория Базовые конструкции TypeScript.

Практика Типизация. Dynamic или Static Typing. Неявные преобразования. Присвоение типов. Настройка TypeScript в проекте.

Тема 21. Основы тестирования (автоматические тесты, Unit-тесты). Отладка, дебаггинг и профилирование приложения

Теория Правила хороших тестов (*ручное и автоматизированное тестирование*).

Практика Тестирование DOM. Заглушка (stub). Test Runner.

Тема 22. Введение в React. JSX. Жизненный циклы компонент (lifecycle)

Теория Основы декларативного подхода. Экосистема React, основные библиотеки. Преимущества JSX.

Практика Настройка редактора VS Code. Настройка плагинов, конфигов и линтеров. Code Style. Особенности синтаксиса JSX. Props в JSX. Дочерние компоненты в JSX.

Тема 23. Функциональный компонент. Hooks

Теория Что такое Hooks. Композиции компонентов. Некоторые правила использования хуков

Практика Props. Состояние в функциональном компоненте. Встроенные хуки (*useState, useEffect, useContext*). Создание пользовательского хука.

Тема 24. PropTypes. Коллекции, обработка событий

Теория Как использовать PropTypes. Основные типы PropTypes. Списки и ключи.

Практика .isRequired, .defaultProps. PropTypes *{объектов}* и *[массивов]*. Передача аргументов в обработчики событий. Способы привязки обработчиков событий.

Тема 25. Styled components. Контекст (Context API). Redux

Теория Мотивация использования Redux. Что такое Context и когда его использовать. Правила работы с Context.

Практика Неизменяемое дерево состояний. Генераторы действий (*actions creators*). Редюсеры. Хранилище. Поток данных. React-redux (*установка пакета*). API (*React.createContext, Context.Consumer, Context.displayName*).

Тема 26. Тестирование. Фреймворк Jest

Теория О фреймворке Jest. Подходы к тестированию React-приложений. Матчеры (matchers) Jest.

Практика Установка и запуск. Настройка приложения и организация тестов. Тестирование с использованием снимков в Jest. Тестирование компонента React.

Тема 27. Основы работы бэкенда

Теория Как устроен интернет. Цикл «запрос – ответ». Как смотреть, что происходит с запросом и ответом в браузере.

Практика HTTP-методы. Коды ответа сервера.

Тема 28. Express.js и базы данных

Теория Переменные окружения. MongoDB. Организация кода.

Практика Установка Express.js и настройка Hot Reload. Роутинг. Middleware.

Тема 29. Rest API

Теория Подходы к разработке API. Принципы REST. Документация API (*OpenAPI*).

Практика RESTful API. Best Practice. Создание тестов для проверки API в Postman.

Тема 30. Node.js

Теория Инструкция по использованию VPN. Отличие между браузерным и серверным JS. Пакет Nodemon.

Практика Импорт и экспорт модулей. Пакет Nodemon (*установка*). Модули (*Console, HTTP, FS*).

Тема 31. Асинхронность. Оптимизация

Теория Асинхронное программирование. Принцип согласования в React / React Reconciliation. Инструменты оптимизации React-приложений.

Практика Redux Thunk (*Установка и настройка*). ShouldComponentUpdate. PureComponent. UseCallBack. Immutable (*иммутабельность данных*).

Тема 32. Формы React Refs

Теория Что такое Refs. Ссылка на классовой компонент.

Практика Создание refs. Доступ к рефам. Добавление ссылки ref на DOM-элемент. ref и функциональный компонент.

Тема 33. Q&A сессия

Теория Инструменты разработки, не вошедшие в данный курс.

Практика Ответы на вопросы учащихся. Оптимизация кода.

Тема 34: Создание творческой итоговая работа.

Финальная работа по созданию приложения

Тема 35: Подготовка к защите проектов.

Подготовка выступлений детей с собственными проектами.

Тема 36: Защита проектов.

Выступление детей с собственными проектами.

Планируемые результаты обучения

Личностные результаты:

- сформированность познавательных интересов, интеллектуальных и творческих способностей учащихся;
- мотивация образовательной деятельности школьников на основе личностно-ориентированного подхода;
- формирование ценностных отношений друг к другу, учителю, авторам открытий и изобретений, результатам обучения.

Метапредметные результаты:

- овладение навыками самостоятельного приобретения новых знаний;

- овладение навыками организации учебной деятельности, постановки целей, планирования, самоконтроля и оценки результатов своей деятельности, умениями предвидеть возможные результаты своих действий;
- формирование умений воспринимать, перерабатывать и предъявлять информацию в словесной, образной, символической формах;
- формирование умений анализировать и перерабатывать полученную информацию в соответствии с поставленными задачами, выделять основное содержание прочитанного текста, находить в нем ответы на поставленные вопросы и излагать его;
- формирование ИКТ-компетентности.

Предметные результаты:

Правила поведения в компьютерном классе.

Учащийся научится:

- соблюдать требования к организации компьютерного и рабочего места;
- выполнять правила поведения в компьютерном классе.

Учащийся получит возможность научиться:

- соблюдать требования безопасности и гигиены в работе со средствами ИКТ.

Учебно-календарный план на 2023/2024 учебный год

Программа: «Веб-разработка»

Педагог дополнительного образования: Курышев Валентин Владимирович

Год обучения: 1 год обучения

<i>№ п/п</i>	<i>Разделы и темы</i>	<i>Кол-во часов по программе</i>	<i>Кол-во часов по факту</i>	<i>Даты занятий</i>
1.	<p>Тема 1. Введение в веб-разработку. Основы HTML и CSS. Строчные и блочные элементы.</p> <p><i>Теория</i> Правила поведения в компьютерном классе. Теория по HTML и CSS.</p> <p><i>Практика</i> Установка на ПК Visual Studio. Добавляем изображения и формы в HTML-страницу. Синтаксис HTML. Список горячих клавиш, которые вам пригодятся.</p>	4 ч		
2.	<p>Тема 2. Позиционирование Flexbox Grid Layout. Препроцессоры CSS. Практика создания адаптивного дизайна.</p> <p><i>Теория</i> Продвинутое свойства стилей. Figma – правила использования. Строчно-блочные элементы. Позиционирование.</p> <p><i>Практика</i> Горизонтальное и вертикальное позиционирование. Центрирование проекта. Примеры позиционирования. Сравнение Flexbox и Grid Layout.</p>	4 ч		
3.	<p>Тема 3. Методология БЭМ и анимация в приложениях</p> <p><i>Теория</i> Методология БЭМ (<i>Как давать логичные имена классам</i>). Проблемы анимации и их решения.</p> <p><i>Практика</i> CSS-переходы (<i>transition</i>). Трансформация (<i>transform</i>). CSS3-анимация.</p>	4 ч		
4.	<p>Тема 4. Знакомство с JavaScript. Условное ветвление и основы функций. Циклы и массивы</p> <p><i>Теория</i> Знакомство с базовыми операторами. Консоль в Google Chrome. Условное ветвление. Встроенные объекты Math().</p> <p><i>Практика</i> Подготовка к созданию JS-кода. Переменные. Методы alert, prompt, confirm. Типы данных в JS. Подготовка рабочей среды</p>	4 ч		

	(установка плагинов). Логические операторы. Реализация циклов while и for.			
5.	<p>Тема 5. Продвинутая работа с циклами и массивами. Работа с объектами. Подготовка к индивидуальному проекту</p> <p><i>Теория</i> Рассмотреть, какие существуют методы работы с массивами, как происходит копирование массива, а также область их применения.</p> <p><i>Практика</i> Методы массива. Очередь и стек. Многомерные массивы. Рекурсия.</p>	4 ч		
6.	<p>Тема 6. Browser Object Model. Введение в Document Object Model</p> <p><i>Теория</i> Детали работы с DOM - как перемещаться по дереву относительно любого произвольного элемента, использование наиболее удобных и продвинутых способов, выбор элементов с помощью знакомых CSS-селекторов.</p> <p><i>Практика</i> Миграция с id на классы. Перемещение по DOM относительно объекта-узла. Коллекции (<i>HtmlCollection, NodeList, Array</i>). Обработчик кликов для нескольких элементов. <i>GetComputedStyle(element)</i>.</p>	4 ч		
7.	<p>Тема 7. Шаблонизация</p> <p><i>Теория</i> Почему лучше не использовать innerHTML. Методы создания верстки в браузере. Добавление элемента в документ.</p> <p><i>Практика</i> Создание объекта-узла. Некоторые приемы при добавлении элементов (<i>в начало и конец списка</i>). Удаление элемента.</p>	4 ч		
8.	<p>Тема 8. События</p> <p><i>Теория</i> Что такое событие и как оно возникает. Как браузер понимает, по какому элементу кликнули. Некоторые популярные события, связанные с мышью, клавиатурой или самим документом. Что такое обработка события браузером по умолчанию. Что такое всплытие и как оно связано с делегированием.</p> <p><i>Практика</i> Способы обработки событий. Основные события мыши. Основные события клавиатуры. Основные события элементов форм. Основные события документа и его элементов. Объект события. Всплытие. Делегирование.</p>	4 ч		

9.	<p>Тема 9. Работа с формами и элементами на странице</p> <p><i>Теория</i> Как форма отправляется по умолчанию. Какие бывают параметры у полей ввода форм и на что они влияют.</p> <p><i>Практика</i> Отправка формы по умолчанию. Action. Передача данных. Типы полей ввода. Основные события для форм (<i>submit, change, input</i>).</p>	4 ч		
10.	<p>Тема 10. Типовые виджеты</p> <p><i>Теория</i> Как разрабатывать комплексные интерфейсные элементы.</p> <p><i>Практика</i> Позиционирование элемента. Перемещение элемента к другому элементу. <code>ElementFromPoint</code>.</p>	4 ч		
11.	<p>Тема 11. Ајах и работа с данными</p> <p><i>Теория</i> Что браузер может запрашивать при переходе по ссылке. В чем разница между асинхронным и синхронным запросами. Почему всегда следует использовать асинхронные запросы.</p> <p><i>Практика</i> Запрос данных браузером. Asynchronous JavaScript And XML. XMLHttpRequest. Обработка ошибок. Живые виджеты. API.</p>	4 ч		
12.	<p>Тема 12. Работа с API</p> <p><i>Теория</i> Что такое API. Как передаются данные в API. Различные способы кодирования данных.</p> <p><i>Практика</i> Передача данных в API (Передача данных через AJAX, с помощью URL, в теле запроса). JSON. CORS. Программы для отправки запросов по HTTP.</p>	4 ч		

13.	<p>Тема 13. Использование сторонних библиотек и организация кода</p> <p><i>Теория</i> Как раньше искали дополнительные библиотеки и виджеты для своего проекта. Какие бывают менеджеры пакетов для Frontend-проектов.</p> <p><i>Практика</i> NPM. Поиск пакета необходимого для проекта. Внедрение найденного пакета.</p>	4 ч		
14.	<p>Тема 14. Ретроспектива JSB</p> <p><i>Теория</i> Структурирование пройденного материала.</p> <p><i>Практика</i> Разбор выполненных домашних заданий.</p>	4 ч		
15.	<p>Тема 15. Q&A сессия</p> <p><i>Теория</i> Инструменты разработки, не вошедшие в данный курс.</p> <p><i>Практика</i> Ответы на вопросы учащихся. Оптимизация кода.</p>	4 ч		
16.	<p>Тема 16. Жизненный цикл проекта и основы GIT</p> <p><i>Теория</i> Жизненный цикл Frontend-проектов. Состав продуктовой команды. Что такое Git.</p> <p><i>Практика</i> SSH-ключи. Commit. Создание репозитория локально. Merge request и Pull Request. Обновление локальной кодовой базы.</p>	4 ч		
17.	<p>Тема 17. Code Review</p> <p><i>Теория</i> Особенности Code Review.</p> <p><i>Практика</i> Создание и принятие Pull Request.</p>	4 ч		
18.	<p>Тема 18. Пакетные менеджеры. Code Linting</p> <p><i>Теория</i> Npm (Nodejs Package Manager). Основные команды для работы. Устройство package.json. Линтеры и формтеры кода.</p> <p><i>Практика</i> Packages. Пакетный менеджер npm. Инициализация фронтového приложения. Стандарты в программировании. Формтеры. Стандарты в программировании. Линтеры.</p>	4 ч		
19.	<p>Тема 19. Инструменты для сборки проектов</p> <p><i>Теория</i> Webpack.</p> <p><i>Практика</i> Подробнее про работу и задачи webpack. Запуск webpack. Плагины (plugins). Оптимизация.</p>	4 ч		

20.	<p>Тема 20. Typescript</p> <p><i>Теория</i> Базовые конструкции TypeScript.</p> <p><i>Практика</i> Типизация. Dynamic или Static Typing. Неявные преобразования. Присвоение типов. Настройка TypeScript в проекте.</p>	4 ч		
21.	<p>Тема 21. Основы тестирования (автоматические тесты, Unit-тесты). Отладка, дебаггинг и профилирование приложения</p> <p><i>Теория</i> Правила хороших тестов (<i>ручное и автоматизированное тестирование</i>).</p> <p><i>Практика</i> Тестирование DOM. Заглушка (stub). Test Runner.</p>	4 ч		
22.	<p>Тема 22. Введение в React. JSX. Жизненный цикл компонент (lifecycle)</p> <p><i>Теория</i> Основы декларативного подхода. Экосистема React, основные библиотеки. Преимущества JSX.</p> <p><i>Практика</i> Настройка редактора VS Code. Настройка плагинов, конфигов и линтеров. Code Style. Особенности синтаксиса JSX. Props в JSX. Дочерние компоненты в JSX.</p>	4 ч		
23.	<p>Тема 23. Функциональный компонент. Hooks</p> <p><i>Теория</i> Что такое Hooks. Композиции компонентов. Некоторые правила использования хуков</p> <p><i>Практика</i> Props. Состояние в функциональном компоненте. Встроенные хуки (<i>useState, useEffect, useContext</i>). Создание пользовательского хука.</p>	4 ч		
24.	<p>Тема 24. PropTypes. Коллекции, обработка событий</p> <p><i>Теория</i> Как использовать PropTypes. Основные типы PropTypes. Списки и ключи.</p> <p><i>Практика</i> .isRequired, .defaultProps. PropTypes {<i>объектов</i>} и [<i>массивов</i>]. Передача аргументов в обработчики событий. Способы привязки обработчиков событий.</p>	4 ч		
25.	<p>Тема 25. Styled components. Контекст (Context API). Redux</p> <p><i>Теория</i> Мотивация использования Redux. Что такое Context и когда его использовать. Правила работы с Context.</p> <p><i>Практика</i> Неизменяемое дерево состояний. Генераторы действий (<i>actions creators</i>).</p>	4 ч		

	Редюсеры. Хранилище. Поток данных. React-redux (<i>установка пакета</i>). API (<i>React.createContext, Context.Consumer, Context.displayName</i>).			
26.	Тема 26. Тестирование. Фреймворк Jest <i>Теория</i> О фреймворке Jest. Подходы к тестированию React-приложений. Матчеры (matchers) Jest. <i>Практика</i> Установка и запуск. Настройка приложения и организация тестов. Тестирование с использованием снимков в Jest. Тестирование компонента React.	4 ч		
27.	Тема 27. Основы работы бэкенда <i>Теория</i> Как устроен интернет. Цикл «запрос – ответ». Как смотреть, что происходит с запросом и ответом в браузере. <i>Практика</i> HTTP-методы. Коды ответа сервера.	4 ч		
28.	Тема 28. Express.js и базы данных <i>Теория</i> Переменные окружения. MongoDB. Организация кода. <i>Практика</i> Установка Express.js и настройка Hot Reload. Роутинг. Middleware.	4 ч		
29.	Тема 29. Rest API <i>Теория</i> Подходы к разработке API. Принципы REST. Документация API (<i>OpenAPI</i>). <i>Практика</i> RESTful API. Best Practice. Создание тестов для проверки API в Postman.	4 ч		
30.	Тема 30. Node.js <i>Теория</i> Инструкция по использованию VPN. Отличие между браузерным и серверным JS. Пакет Nodemon. <i>Практика</i> Импорт и экспорт модулей. Пакет Nodemon (<i>установка</i>). Модули (<i>Console, HTTP, FS</i>).	4 ч		
31.	Тема 31. Асинхронность. Оптимизация <i>Теория</i> Асинхронное программирование. Принцип согласования в React / React Reconciliation. Инструменты оптимизации React-приложений. <i>Практика</i> Redux Thunk (<i>Установка и настройка</i>). ShouldComponentUpdate. PureComponent. UseCallBack. Immutable (<i>иммутабельность данных</i>).	4 ч		

32.	<p>Тема 32. Формы React Refs</p> <p><i>Теория</i> Что такое Refs. Ссылка на классовой компонент.</p> <p><i>Практика</i> Создание refs. Доступ к рефам. Добавление ссылки ref на DOM-элемент. Ref и функциональный компонент.</p>	4 ч		
33.	<p>Тема 33. Q&A сессия</p> <p><i>Теория</i> Инструменты разработки, не вошедшие в данный курс.</p> <p><i>Практика</i> Ответы на вопросы учащихся. Оптимизация кода.</p>	4 ч		
34.	<p>Тема 34: Создание творческая итоговая работа.</p> <p>Финальная работа по созданию приложения</p>	4 ч		
35.	<p>Тема 35: Подготовка к защите проектов.</p> <p>Подготовка выступлений детей с собственными проектами.</p>	4 ч		
36.	<p>Тема 35: Защита проектов.</p> <p>Выступление детей с собственными проектами</p>	4 ч		
	Итого:	144		

Список литературы для педагогов

1. Morgan J. How To Code in React.js. DigitalOcean. 2021
2. Полный курс по React. [Электронный ресурс]. URL: <https://habr.com/ru/company/ruvds/blog/471040/> (дата обращения: 01.09.2023).
3. Хавербеке М. Выразительный JavaScript. Современное веб-программирование. Питер. 2019
4. Фаулер М. Рефакторинг кода на JavaScript. Вильямс. 2019
5. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. Питер. 2019
6. Прохоренок Н.А., Дронов В.А. JavaScript и Node.js для веб-разработчиков. БХВ. 2022

Список литературы для детей

1. Морган Н. JavaScript для детей. Самоучитель по программированию. Манн. 2018
2. Флэнаган Д. JavaScript. Подробное руководство. 6-е издание. Символ-Плюс. 2021
3. Эрик Фримен, Элизабет Робсон. Изучаем программирование на JavaScript. Питер. 2021
4. Справочник по HTML и CSS. [Электронный ресурс]. URL: <http://htmlbook.ru/> (дата обращения: 07.09.2022).
5. Изучаем HTML. [Электронный ресурс]. URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics (дата обращения: 07.09.2022).
6. Морган Н. Самоучитель по JavaScript для детей. Манн. 2019

Материалы по отслеживанию результативности образовательной деятельности

Критерии результативности и диагностические материалы для определения результатов и качества образовательного процесса представлены в *Приложение № 1*.

Для контроля результативности обучения используются **карты фиксации образовательных результатов: входная, промежуточная и итоговая** (форма карты в *Приложение № 2*) и **таблица фиксации образовательных результатов** проведенных мероприятий (*Приложение № 3*).

Критерии результативности и диагностические материалы для определения результатов и качества образовательного процесса.

Критерии результативности:

- Степень владения компьютером и другими техническими средствами, необходимыми для разработки приложения;
- Умение самостоятельно находить необходимую информацию;
- Умение программировать по шаблону;
- Умение самостоятельно программировать;
- Умение работать в браузерах;
- Знание и умение использовать алгоритмы, массивы, циклы и простейшие функции;
- Умение работать в паре и коллективе;
- Умение связно рассказать о своем коде: для чего, и какие технологии были задействованы, по какому принципу работает, продемонстрировать работу;
- Умение организовывать рабочее место.

Система оценивания – 10-ти бальная:

- «10» — очень хороший, высокий уровень знаний, полное владение навыком
- «9» — высокий уровень знаний, владение навыком достаточное
- «8» — «крепкие» знания, иногда очень хорошие, владение навыком достаточное
- «7» — «крепкие» знания, достаточное владение навыком
- «6» — «крепкие» знания, но есть некоторые недоработки в практической работе
- «5» — уровень знаний средне-невысокий, частичное владение навыком
- «4» — «слабый» уровень знаний, частичное владение навыком
- «3» — «слабый» уровень знаний, навык практической работы минимален
- «2» — «слабый» уровень знаний, навык практической работы отсутствует
- «1» — знания отсутствуют, навык практической работы отсутствует

Карта фиксации образовательных результатов
Программа «*Веб-разработка*»
Курышев Валентин Владимирович
2023 / 2024 учебный год

	Умения самостоятельно находить необходимую информацию	Умение работать в программах и работать по шаблону	Умение работать в браузерах	Умение работать в паре и коллективе	Умение работать в паре и коллективе	Умение работать в паре и коллективе	Умение организовывать рабочее место	%
--	---	--	-----------------------------	-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	---

Max	10	10	10	10	10	10	10	

Приложение № 3

Таблица фиксации образовательных результатов проведенных мероприятий
ОДОД ГБОУ школа № 525
2023 / 2024 учебный год

п/п	Мероприятие	Место	Время проведения	Результат
1				
2				
3				

п/п	Мероприятие	Место	Время проведения	Результат
4				
...				