

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ ГОРОДА КАЛУГИ

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ «ДЕТСКО-ЮНОШЕСКИЙ ЦЕНТР
КОСМИЧЕСКОГО ОБРАЗОВАНИЯ «ГАЛАКТИКА» ГОРОДА КАЛУГИ

ПРИНЯТА
педагогическим советом
МБОУДО ДЮЦКО
«Галактика» г. Калуги
Протокол № 1 от 30.08.2023



УТВЕРЖДАЮ
директор МБОУДО ДЮЦКО
«Галактика» г. Калуги
Приказ № 266/01-09 от 31.08.2023
А.Ю. Кононова



IT-КВАНТУМ

Дополнительная общеобразовательная общеразвивающая программа
технической направленности
детского технопарка «Кванториум»

Команда JavaScript

Срок реализации программы – 1 год
Возраст: 12-17 лет

Автор-составитель:
Купцов Александр Борисович,
педагог дополнительного образования

Калуга, 2023

Паспорт программы

Полное название программы	Дополнительная общеобразовательная общеразвивающая программа технической направленности «Команда JavaScript»
Автор-составитель программы	Купцов Александр Борисович, педагог дополнительного образования
Адрес реализации программы	Муниципальное бюджетное образовательное учреждение дополнительного образования «Детско-юношеский центр космического образования «Галактика» города Калуги, НСП «Детский технопарк «Кванториум» 248 002, г. Калуга, ул. С. Щедрина, д. 66, тел. 8 (4842) 79 74 90
Вид программы	- по степени авторства – модифицированная; - по уровню сложности – продвинутый
Направленность программы	Техническая
Срок реализации программы	1 год, 144 часа в год
Возраст обучающихся	12-17 лет
Название объединения	Команда JavaScript

1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Команда JavaScript» направлена на приобретение учащимися навыков использования информационных технологий в учебной и повседневной деятельности; на формирование навыков алгоритмизации и программирования, развитие инженерного мышления, формирование профессиональной направленности.

Направленность программы – техническая.

Вид программы

По степени авторства – модифицированная.

По уровню сложности – продвинутый

Язык реализации программы - русский.

Дополнительная общеобразовательная программа разработана в соответствии с нормативно-правовыми документами:

- Федеральным Законом РФ от 29.12.2012 № 273 «Об образовании в Российской Федерации»;

- Приказом Министерства просвещения РФ от 27.07.2022 № 629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

- Распоряжением Правительства Российской Федерации от 31.03.2022 № 678-р «Концепция развития дополнительного образования детей до 2030 года»;

- Распоряжением Правительства Российской Федерации от 29.05.2015 № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;

- Постановлением Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СанПиН 2.4.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

- Письмом Министерства образования и науки РФ от 18.11.2015 № 09-3242 «О направлении информации» с методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)».

Актуальность, отличительные особенности

Для эффективного взаимодействия с окружающей действительностью в информационном обществе важно ещё в школе овладеть информационными и коммуникативными технологиями. Настоящая программа позволит школьникам сделать первые шаги к профессиям фронтенд и бекенд-разработчики. Осваивая материал программы учащиеся получают навыки решения типовых задач, научатся выполнять настоящие проекты, такие же, как и в веб индустрии. Поэтому очень важно ещё в школе овладеть навыками веб-разработчика, и чем раньше, тем больший запас знаний и технологий получит ребенок к моменту выбора профессии. Даже если в будущем карьерный путь обучающегося не будет связан с веб-разработкой, умение разбираться в сложных системах и взаимодействовать с новыми технологиями в будущем ему пригодится в любой сфере.

Программу можно рассматривать как глубокое знакомство с профессией веб-разработчика, позволяющее сформировать у слушателей представление о своей будущей специальности.

В программе используется курс «Команда JavaScript», созданный благотворительным фондом развития образования «Айкью Опшн» («Возможность Интеллекта»), который реализует проект совместно с HTML Academy при участии Geek Teachers. Каждый учащийся будет иметь доступ к личному онлайн - кабинету на сайте htmlacademy.ru, где он сможет изучать

теоретическую часть и выполнять практические задания по данной программе, а у педагога есть возможность отслеживания процесс обучения учащихся.

Новизна

Программа построена в соответствии с требованиями современного общества к образованию: обеспечение самоопределения личности, создание условий развития мотивации ребёнка к познанию и творчеству, использует принципы вытягивающей модели обучения, создание условий для самореализации, оказание помощи найти своё место в современном информационном мире.

Педагогическая целесообразность состоит в том, чтобы сформировать у подрастающего поколения новые компетенции, необходимые в обществе, использующем современные информационные технологии; позволяющие обеспечивать динамическое развитие личности ребенка, его нравственное становление; формировать целостное восприятие мира, людей и самого себя, развивать интеллектуальные и творческие способности ребенка. По мере изучения программирования, у обучающихся формируется не только логическое мышление, но и навыки работы с мультимедиа, создаются условия для активного, поискового учения, разнообразного программирования. Индивидуальная работа над проектом развивает самостоятельность, привносит соревновательный элемент, позволяет наиболее полно раскрыть интеллектуальный потенциал учащегося.

Программа направлена на выявление и развитие одаренных детей. Развивается умение выделять систему понятий, представлять их в виде совокупности атрибутов и действий, описать алгоритмы действий и схемы логического вывода (то есть то, что и происходит при информационно-логическом моделировании), занятия улучшают ориентацию ребенка в любой предметной области.

Всё это решающим образом сказывается на формировании и закреплении новой системы отношений к людям, коллективу, к учению и связанным с ними обязанностям, формирует характер, волю, расширяет круг интересов, развивает способности.

Адресат программы – учащиеся детского технопарка «Кванториум» в возрасте 12-17 лет. Программа построена с учётом возрастных и психологических особенностей учащихся, реализуется для всех желающих в разновозрастных группах; рассчитана на любой социальный статус учащихся, имеющих различные интеллектуальные, технические, творческие способности. На начальной стадии обучения школьникам необходимы знания основ HTML, CSS. обладать достаточной компьютерной грамотностью, чтобы устанавливать и запускать программы.

Набор в группы проводится без предварительного отбора. Комплектование групп проходит с учетом индивидуальных особенностей учащихся.

Количество учащихся в группе – до 15 человек.

Получение образования обучающихся с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися. Количество обучающихся с ограниченными возможностями здоровья устанавливается из расчета не более 3 обучающихся при получении образования с другими учащимися.

Организацию работы, порядок деятельности, продолжительность учебных занятий, количество обучающихся в детских творческих объединениях МБОУДО ДЮЦКО «Галактика» города Калуги регулирует «Положение о детском творческом объединении», утвержденное приказом директора № 122/-09 от 15.08.2022.

Объем программы – 1 год обучения - 144 часа в год;

Срок освоения программы – 1 год,

Уровень сложности - углубленный модуль – «продвинутый».

Формы обучения – индивидуальная, групповая, подгрупповая.

Форма обучения – очная, возможно применение дистанционных технологий.

Формы проведения аудиторных занятий утверждены локальным нормативным актом - «Положение о детском творческом объединении» (приказ директора МБОУДО ДЮЦКО «Галактика» г. Калуги № 122/-09 от 15.08.2022).

Виды занятий - изучение программы предусматривает проведение теоретических и практических занятий.

Режим занятий – занятия в объединении проводятся - 2 раза в неделю по 2 академических часа.

Продолжительность одного занятия составляет 45 минут.

Расписание занятий формируется по представлению педагога с учетом пожеланий обучающихся, родителей (законных представителей) несовершеннолетних обучающихся и возрастных особенностей учащихся.

1.2. Цель и задачи дополнительной программы

Цель – обучение программированию посредством языка JavaScript, развитие инженерного мышления, формирование профессиональной направленности.

Для достижения цели необходимо решить следующие **задачи**:

Образовательные задачи:

1. Обучение базовым навыкам использования компьютера как рабочего инструмента, усвоение соответствующих правил техники безопасности;
2. Научить обучающегося применять базовые понятия программирования: переменная, условия, циклы, функция и т.д. для написания интерактивных элементов сайта;
3. Научить обучающегося применять принципы функционирования JavaScript в браузере и управлять элементами DOM;
4. Сложить для учащихся целостное представление о технологической цепочке создания web-сайтов и сформировать понимание актуальных тенденций развития webтехнологий;
5. Формирование первоначального понимания основ программирования и реализации анимации, образовательных проектов и компьютерных игр;
6. Изучить базовые концепции языка Javascript;
7. Научить делать браузерные интерфейсы;
8. Научить тестировать и проверять разработанные приложения.

Развивающие задачи:

1. Развитие произвольности психических процессов, абстрактно - логических и наглядно - образных видов мышления и типов памяти, основных мыслительных операций, основных свойств внимания;
2. Совершенствование диалогической речи детей: умение слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них;
3. Развитие творческого и рационального подхода к решению поставленных задач;
4. Развитие мелкой моторики, зрительного восприятия, через компьютерные задания, игры, тренажеры.

Воспитательные задачи:

1. Формирование информационной культуры;
2. Воспитание умения работать в мини-группе, культуру общения, ведение диалога;
3. Воспитание настойчивости, собранности, организованности, аккуратности;
4. Воспитание бережного отношения к имуществу;
5. Формирование навыков здорового образа жизни посредством осознания правил безопасной работы с компьютером.

1.3. Содержание программы

Учебный план, 144 часа в год
Углубленный модуль

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Вводное занятие. Правила безопасности	2	2	0	Устный опрос
Раздел 2 «Команда JS»		38	19	19	
2.1	Подготовка к работе. Установка программного обеспечения. Установка и настройка среды разработки.	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.2	Знакомство с JavaScript	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.3	Логический тип данных. Условия	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.4	Циклы	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.5	Массивы	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.6	Функции	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.7	Объекты	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.8	Знакомство с проектом	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.9	DOM	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.10	События и слушатель/обработчик событий	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.

2.11	События клика и смены переключателя	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.12	Хранение данных с помощью <code>template</code> , клонирование и вставка элементов на страницу	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.13	Работа над проектом: добавление элементов	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.14	Удаление и редактирование элементов	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.15	Редактирование текста, замена картинок	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.16	Подготовка проекта	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.17	Подготовка презентации проекта. Турнир.	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.18	Публикация сайта	2	1	1	Практическая работа. Просмотр. Обсуждение. Анализ работы. Тест.
2.19	Защита проекта, подведение итогов	2	2	-	Обсуждение. Анализ работы. Демонстрация и защита проекта
Раздел 3 JavaScript в микроконтроллере		48	14	34	
3.1	JavaScript в микроконтроллере Искра JS	20	6	14	Практическая работа. Просмотр. Обсуждение. Анализ работы.
3.2	Проект «Мой артефакт на JS»	28	8	20	Практическая работа. Просмотр. Обсуждение. Анализ работы. Демонстрация и защита проекта
Раздел 4 Проектная деятельность. Платформа Node.js. Фреймворки Flask, Vue.		56	23	33	

4.1	Node JS, Express JS	4	2	2	Практическая работа. Просмотр. Обсуждение. Анализ работы
4.2	Vue JS	4	2	2	Практическая работа. Просмотр. Обсуждение. Анализ работы.
4.3	Кейс «Прогноз погоды по API»	8	2	6	Практическая работа. Просмотр. Обсуждение. Анализ работы.
4.4	Кейс «Создание сайта-портфолио с помощью Flask»	10	4	6	Практическая работа. Просмотр. Обсуждение. Анализ работы.
4.5	Выполнение проекта с использованием изученных фреймворков.	28	12	16	Практическая работа. Обсуждение. Анализ работы. Демонстрация и защита проекта
5	Заключительное занятие	2	1	1	Тестирование, рефлексия. Анализ работы
Итого		144	55	89	

**Содержание учебного плана, 144 часа в год
Углубленный модуль**

1. Вводное занятие. Правила безопасности (2 ч.).

Теория (2 ч.). Правила поведения в компьютерном классе. Электробезопасность. Противопожарные мероприятия.

2. «Команда JS» (2 ч.)

2.1 Подготовка к работе. Установка программного обеспечения. Установка и настройка среды разработки (38 ч.).

Теория (1 ч.) Знакомство с средой разработки. Visual studio code. Плагины.

Практика (1 ч.). Установка программного обеспечения. Установка и настройка среды разработки (1 ч.).

2.2 Знакомство с JavaScript (2 ч.).

Теория (1 ч.) Что такое JavaScript. что такое алгоритм и как «думает» компьютер. Понятие переменных и консоли.

Практика (1 ч.). Изучение инструментов веб разработки. Регистрация на платформе «HTML Academy» и выполнение практической работы

2.3 Логический тип данных. Условия (2 ч.).

Теория (1 ч.) Условия в программе. Приведение типов. Функции взаимодействия с пользователем

Практика (1 ч.). Попытки написать первый код. Практическая работа с кодом, решение задачи “Неприличный вопрос”

2.4 Циклы (2 ч.).

Теория (1 ч.) Циклические алгоритмы while, do while, for

Практика (1 ч.). Практические упражнения на применение циклов.

2.5 Массивы (2 ч.).

Теория (1 ч.) Понятие массива. Использование данных массива

Практика (1 ч.). Практическая работа по написанию кода. Выполнение задания на платформе Академии

2.6 Функции (2 ч.).

Теория (1 ч.) Понятие функции. Параметры функции. Метод `forEach` для работы с массивами.

Практика (1 ч.). Практическая работа по написанию кода. Выполнение задания Демо на платформе Академии

2.7 Объекты (2 ч.)

Теория (1 ч.) Понятие объекта. Получение данных. Запись в объект.

Практика (1 ч.). Выполнение задания в LearningApps и на платформе Академии

2.8 Знакомство с проектом (2 ч.)

Теория (1 ч.) Знакомство с проектом и ТЗ. Информация по верстке проекта.

Практика (1 ч.). Использование средств разработчика в браузере Google Chrome. Практическая работа по написанию кода.

2.9 DOM (2 ч.)

Теория (1 ч.) Демо по теме урока. Работа на создание дерева элементов.

Практика (1 ч.). Нахождение элементов страницы через консоль. Выполнение задания на платформе Академии

2.10 События и слушатель/обработчик событий (2 ч.)

Теория (1 ч.) Знакомство с событиями и слушателем событий, запись обработчика событий. Демо на цикл `forEach`

Практика (1 ч.). Практическая работа по написанию кода.

2.11 События клика и смены переключателя (2 ч.)

Теория (1 ч.) Переключатели и событие `change`. Получение данных из `value`.

Практика (1 ч.). Работа с Демо. Практическая работа по написанию кода.

2.12 Хранение данных с помощью `template`, клонирование и вставка элементов на страницу (2 ч.)

Теория (1 ч.) Шаблоны `template`. Клонирование элементов. Вставка элементов на страницу

Практика (1 ч.). Работа с Демо. Практическая работа по написанию кода.

2.13 Работа над проектом: добавление элементов (2 ч.)

Теория (1 ч.) Data-атрибуты. Игра. Создай свой макет и проверь других. Игра на запоминание

Практика (1 ч.). Работа с Демо. Практическая работа по написанию кода

2.14 Удаление и редактирование элементов (2 ч.)

Теория (1 ч.) Разминка “Угадай”. Демо по теме урока. Видео-рассказ тестировщика о своей профессии

Практика (1 ч.). Работа с Демо. Практическая работа по написанию кода

2.15 Редактирование текста, замена картинок (2 ч.)

Теория (1 ч.) Демо по теме урока. Командная работа по исследованию кода. Стена впечатлений и благодарностей

Практика (1 ч.). Работа с Демо. Практическая работа по написанию кода

2.16 Подготовка проекта (2 ч.)

Теория (1 ч.) Повторение в виде игры в LearningApps. Минимизация файлов. Создание скриншотов.

Практика (1 ч.). Отладка кода проекта. Подготовка проекта к защите.

2.17 Подготовка презентации проекта. Турнир (2 ч.)

Теория (1 ч.) Создание красивой презентации. Возможности применения JavaScript на примере создания игр.

Практика (1 ч.). Турнир по игре, написанной на JavaScript

2.18 Публикация сайта (2 ч.)

Теория (1 ч.) Знакомство с понятиями Домен и хостинг. Подготовка к выступлению на уроке

Практика (1 ч.). Знакомство и регистрация в gitHub. Публикация сайта

2.19 Защита проекта, подведение итогов (2 ч.)

Теория (1 ч.) Защита проекта

Практика (1 ч.). Работа с Демо.

3. JavaScript в микроконтроллере (48 ч.).

3.1 JavaScript в микроконтроллере Искра JS (20 ч.)

Теория (6 ч.). Начало работы с JavaScript-контроллерами. Среда программирования Espruino Web IDE. Знакомство с набором ЙОДО. IoT (Интернет вещей) — дополнение к набору «Йодо»

Практика (14 ч.). Эксперименты: 1. Удалённый термометр, Восьмибитный аудиоплеер, Браузерный Dendy, Умный дом, Telegram bot

3.2 Проект «Мой артефакт на JS» (28 ч.).

Разработка и реализация проекта с использованием языка программирования JavaScript и микроконтроллера Iskra JS. Защита проектов.

4. Проектная деятельность. Платформа Node js. Фреймворки Flask, Vue (56 ч.)

4.1 Node JS, Express JS (4 ч.)

Теория (2 ч.). Изучение платформы Node JS, Express JS и разработка сайта на их основе

Практика (2 ч.). Установка. Настройка. Разработка веб сайта

4.2 Vue JS (4 ч.).

Теория (2 ч.). Знакомство с Vue JS. Разработка проектов на его основе

Практика (2 ч.). Настройка проекта. Структура. Работа с проектом

4.3 Кейс «Прогноз погоды по API» (8 ч.)

Теория (2 ч.). Познакомиться с API, выяснить что они могут делать и как их использовать. Рассматриваем на примере API сайта OpenWeatherMap.org.

Практика (6 ч.). Регистрации на сайте OpenWeatherMap.org. Сформируем и отправим запрос, разберем ответный пакет в формате JSON, и получим текущую температуру с описанием состояния погоды.

4.4 Кейс «Создание сайта-портфолио с помощью Flask» (10 ч.)

Теория (4 ч.). Знакомство с фреймворком Flask и SQLite в Python 3.

Практика (6 ч.). Учащиеся создадут небольшой сайт-портфолио с использованием фреймворка Flask и SQLite в Python 3. На сайте будет размещена информация об учащимся. Примеры его работ. Подключена база данных для создания, редактирования и удаления постов.

4.5 Выполнение проекта с использованием изученных фреймворков. (28 ч.)

Разработка и реализация проекта с использованием языка программирования JavaScript и изученных фреймворков. (Веб приложение, сайт, игра, тест). Размещение проектов в сети. Защита проектов.

5. Заключительное занятие (2 ч.).

Анализ работы за год. Подведение итогов. Задание на лето.

1.4. Планируемые результаты

Учащиеся, освоившие программу должны:

Знать:

- как самостоятельно добывать и обрабатывать информацию по предмету, получить навыки дальнейшего самостоятельного обучения и самообразования;
- как ориентироваться в современном информационном обществе и освоит новейшие методы образования: обучающие программы, электронные документы, развивающие игры и т.д.;
- как получить культурологические и экологические знания. Приобщатся к культурному наследию. Все это, в конечном итоге будет способствовать формированию гражданской ответственности подрастающего поколения;
- алгоритмические конструкции;
- общую структуру языка JavaScript, его синтаксис;
- знать и уметь применять базовые понятия программирования: переменная, условия, циклы, функция и т.д. для написания интерактивных элементов сайта;

- знать и уметь применять принципы функционирования JavaScript в браузере и уметь управлять его элементами;
- уметь дорабатывать код верстки для программ на JavaScript;
- уметь "оживлять" интерфейс веб-страницы и работать с пользовательскими событиями;
- знать устройство клиент-серверного взаимодействия и уметь работать с системами контроля версий (Git) на базовом уровне: создание репозитория, загрузка файлов, работа с сервисом GitHub Pages;
- уметь выстроить работу над проектом по алгоритму;
- владеть навыками проверки и отладки кода web-документов.

Уметь:

- верстать HTML-страницы, задавать свойства CSS;
- интегрировать сценарий JavaScript в код HTML-страницы;
- изменять содержимое HTML-страниц средствами JavaScript, в том числе добавлять, удалять, отображать и скрывать элементы объектной модели документа, вносить изменения в их текстовое наполнение;
- модифицировать оформление HTML-страниц средствами JavaScript, в том числе форматировать текст, списки, таблицы, управлять положением блоков на странице, добавлять элементам классы CSS, вносить изменения в свойства CSS;
- создавать интерактивные элементы интерфейса веб-страниц, такие как меню, вкладки, ролловеры, галереи изображений;
- создавать анимированные приложения средствами JavaScript;
- использовать ветвления и циклы различного вида для создания многоуровневых квестов и интерактивных историй с ветвящимися сюжетами;
- использовать компьютер для решения учебных и простейших практических задач, для этого: иметь начальные навыки использования компьютерной техники, уметь осуществлять простейшие операции с файлами (создание, сохранение, поиск, запуск программы); запускать простейшие широко используемые прикладные программы: текстовый и графический редактор, тренажёры и тесты;
- планировать и создавать творческие проекты в среде программирования Espruino Web IDE, JavaScript.

Обладать

- навыками организации индивидуального информационного пространства, создания личных коллекций информационных объектов;
- опытом использования приобретенных знаний и умений в практической деятельности и в повседневной жизни;
- приемами проектирования интерфейса проекта;
- приемами выбора метода анимации для конкретной задачи;
- приемами планирования последовательности событий для создания эффекта анимации по выбранному сценарию.

Прогнозируемые результаты

Учебные действия	Учащиеся знают, понимают	Учащиеся умеют
Регулятивные	<ul style="list-style-type: none"> • Требования правил по технике безопасности 	<ul style="list-style-type: none"> • Самостоятельно добывать и обрабатывать информацию по предмету • Имеют навыки дальнейшего самостоятельного обучения и самообразования

<p>Познавательные</p>	<ul style="list-style-type: none"> • Источники информации (книги, пресса, радио и телевидение, Интернет, устные сообщения) • Ориентироваться в современном информационном обществе и освоит новейшие методы образования: обучающие программы, электронные документы, развивающие игры • Знания для оформления в текстовом и графическом редакторе любой информации • Основы HTML и CSS • Основы языка программирования Javascript. • Основы работы в программных средах VS, Espruino Web IDE 	<ul style="list-style-type: none"> • Внедрять код в Html страницы; • Применять основы HTML-теги, атрибуты и способы структурирования содержимого web-страниц для создания форматированных документов; • Применять основы CSS-значения, списки, цвета, шрифты и другие метрики форматирования; • Владеть навыками проверки и отладки кода web-документов; • Владеть навыками быстрого и качественного форматирования сложных web-документов; • Создавать динамических веб-страниц средствами Java Script. • Создавать программы на языке JavaScript; • Инициировать и вести творческие проекты
<p>Личностные</p>	<ul style="list-style-type: none"> • Способы применения полученных знаний и умений на практике • Этические нормы при работе с информацией • Культурологические и экологические знания, способствующие формированию активной гражданской позиции 	<ul style="list-style-type: none"> • Определять свое эмоциональное отношение к работе и получаемым результатам • Ориентироваться в современном информационном обществе и освоит новейшие методы образования: обучающие программы, электронные документы, развивающие игры •
<p>Коммуникативные</p>	<ul style="list-style-type: none"> • Правила и приемы выполнения групповых творческих работ (микрогруппы) 	<ul style="list-style-type: none"> • Работать в коллективе и в паре • Договариваться в ходе выполнения групповых заданий • сочетать свои интересы с интересами коллектива

2.Комплекс организационно-педагогических условий

2.1. Календарный учебный график (см. рабочую программу).

Календарный учебный график составляется педагогом на основании реализуемой общеобразовательной программы до начала учебного года или начала реализации программы. Календарный учебный график разрабатывается педагогом для каждой группы в форме таблицы, представленной ниже.

№ п/п	Месяц	Число	Время проведения занятия	Форма занятий	Кол-во часов	Тема занятий	Место проведения	Форма контроля

2.2. Условия реализации программы

Материально-техническое обеспечение программы:

- Компьютер;
- Проектор;
- Экран;
- Принтер;
- Роутер;
- Устройства вывода звуковой информации;
- Устройства для ручного ввода текстовой информации и манипулирования экранными объектами — клавиатура и мышь;
- Устройства создания графической информации (графический планшет);
- Устройства для записи (ввода) визуальной и звуковой информации: сканер; фотоаппарат; видеокамера;
- Наборы Искра JS (Йода и Умный дом) Программные средства;
- Операционная система;
- VS, Espruino Web IDE;
- Клавиатурный тренажер;
- Интегрированное офисное приложение, включающее текстовый редактор, растровый и векторный графические редакторы, программу разработки презентаций и электронные таблицы;
- Звуковой редактор;
- Система оптического распознавания текста;
- Мультимедиа проигрыватель (входит в состав операционных систем или др.);
- Почтовый клиент (входит в состав операционных систем или др.);
- Браузер (входит в состав операционных систем или др.).

1.3. Формы аттестации (контроля)

На протяжении всего учебного процесса предлагается проводить следующие виды контроля знаний и аттестации учащихся:

Тестирование;

Участие в городских и областных и международных конкурсах по информатике; Создание проекта по образцу и создание своего проекта по завершении всего курса.

Аттестация проводится в форме выполнения индивидуальных и групповых заданий по пройденному материалу. Контроль в указанной форме осуществляется как промежуточный, так и итоговый. Отметочная форма контроля отсутствуют. Оценка производится на основе критериального оценивания. По итогам работы над групповыми и индивидуальными проектами проводится обсуждение результатов в коллективе с опорой на чек лист, исправление ошибок и, тем самым, коррекция и закрепление полученных знаний. Презентация проекта: учащийся демонстрирует свой проект всему классу и педагогу на занятии, отвечает на вопросы учеников и педагога; педагог акцентирует внимание на сильных сторонах проекта, оценивает техническую сторону исполнения, затем анализирует недочеты, указывает на причины их возникновения; высказывает рекомендации по доработке проекта.

Данная общеобразовательная программа не предусматривает выдачу документа об обучении.

3.4. Оценочные материалы

Система психолого-педагогического мониторинга эффективности реализации образовательной программы

Этап аттестации	Педагогические задачи	Категория испытуемых	Параметры диагностики	Используемые методики	Сроки проведения
Первичная	обучающие	Учащиеся	Начальный уровень учебной подготовки, уровень интеллектуального развития	1. Авторский опросник уровня теоретической и практической подготовки 2. Тест структуры интеллекта Амтхауэра	Сентябрь-октябрь
		Учащиеся с 12 лет			
	развивающие	Учащиеся	Творческие способности	1. Краткий тест творческих способностей (модификация теста Торренса); опросник 2. «Диагностика личностной креативности» (Е.Е.Туник). 3. Диагностика вербальной креативности (методика С.Медника, адаптирована А.Н.Ворониным, 1994)	Октябрь
Учащиеся с 12 лет					
Промежуточная	воспитательные	Учащиеся	Уровень самооценки; преобладающая мотивация (достижения успехов или избегания неудач)	1. Опросник «Шкала оценки потребности в достижении» 2. Опросник оценки уровня уверенности в себе	Октябрь
		Родители, учащиеся	Социальный заказ к учреждению	1. Анкетирование (авторская анкета); 2. анкета «Мои интересы»	Сентябрь
	Учащиеся	Направленность личности	Методика «Выбор»	Сентябрь	
Промежуточная	воспитательные	Учащиеся	Атмосфера в группе	1. Методика «Атмосфера в группе» 2. Индекс групповой сплоченности Сижора	Апрель

	обучающие	Учащиеся	Промежуточный уровень учебной подготовки,	Авторский опросник уровня теоретической и практической подготовки	Апрель
Промежуточная	развивающие	Учащиеся	Профессиональная направленность личности	Методика «Мотивы выбора профессии»; Опросник профессиональных склонностей Л. Йовайши (модификация Г.В. Резапкиной); Тест «Конструктивный рисунок человека из геометрических фигур»	Январь
Итоговая	обучающие	Учащиеся с 12 лет	Итоговый уровень учебной подготовки, уровень интеллектуального развития	1. Авторский опросник уровня теоретической подготовки 2. Тест структуры интеллекта Амтхауэра	Май (в конце обучения)
	развивающие	Учащиеся	Творческие способности	1. Краткий тест творческих способностей (модификация теста Торренса); опросник « <u>Диагностика личностной креативности</u> » (Е.Е. Туник). 3. Диагностика вербальной креативности (методика С.Медника, адаптирована А.Н. Ворониным, 1994) 4. Анализ продуктов творческой деятельности: презентации работ, участие в городских и региональных конкурсах; метод наблюдений; метод экспертных оценок	Май (в конце курса) В ходе обучения
		Учащиеся	Уровень самооценки; преобладающая мотивация (достижения успехов или избегания неудач)	1. Опросник «Шкала оценки потребности в достижении»; 2. Опросник оценки уровня уверенности в себе	Апрель (в конце курса)

Итоговая	воспитательные	Родители, учащиеся	Удовлетворенность результатам и учебно-воспитательной работы	Анкетирование (авторские анкеты)	Апрель (в конце курса)
		Учащиеся	Атмосфера в группе	1. Методика «Атмосфера в группе» 2. Индекс групповой сплоченности Сижора	Апрель (в конце курса)
		Учащиеся	Направленность личности	Методика «Выбор»	Апрель (в конце курса)

Критерии оценки – проекта

№	Критерий	Оценка (в баллах)
1	Актуальность поставленной задачи	3 – имеет большой интерес (интересная тема) 2 – носит вспомогательный характер 1 – степень актуальности определить сложно 0 – не актуальна
2	Новизна решаемой задачи	3 – поставлена новая задача 2 – решение данной задачи рассмотрено с новой точки зрения, новыми методами 1 – задача имеет элемент новизны 0 – задача известна давно
3	Оригинальность методов решения задачи	3 – задача решена новыми оригинальными методами 2 – использование нового подхода к решению идеи 1 – используются традиционные методы решения
4	Практическое значение результатов работы	2 – результаты заслуживают практического использования 1 – можно использовать в учебном процессе 0 – не заслуживают внимания

3.5. Методическое обеспечение программы

Методы и методические приемы:

Занятие – игра. Учащиеся в игровой форме работают с исполнителем, задают ему команды, которые он должен выполнить и достичь поставленной цели (используются различные игры: на развитие внимания и закрепления терминологии, игры-тренинги, игры конкурсы, сюжетные игры на закрепление пройденного материала, интеллектуально познавательные игры, интеллектуально-творческие игры).

Занятие – исследование. Обучающимся предлагается создать рисунки в векторном и растровом редакторах и провести ряд действий, после чего заполнить таблицу своих наблюдений.

Детям предлагается создать рисунок в растровом редакторе и сохранить его с разным расширением, посмотреть, что изменилось, выводы записать на листок.

Практикум – это общее задание для всех учащихся группы, выполняемое на компьютере.

Занятие – беседа. Ведется диалог между педагогом и ребенком, что позволяет обучающимся быть полноценными участниками занятия.

Индивидуальные практические работы - мини-проекты.

Заключительное занятие, завершающее тему – защита проекта. Проводится для самих детей, педагогов, родителей.

Программа предусматривает использование следующих форм работы:

- фронтальной - подача материала всему детскому коллективу
- индивидуальной - самостоятельная работа обучающихся с оказанием педагогом помощи детям при возникновении затруднения, не уменьшая активности обучающихся и содействуя выработке навыков самостоятельной работы.

- групповой - когда обучающимся предоставляется возможность самостоятельно построить свою деятельность на основе принципа взаимозаменяемости, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности. Всё это способствует более быстрому и качественному выполнению заданий. Особым приёмом при организации групповой формы работы является ориентирование детей на создание так называемых минигрупп или подгрупп с учётом их возраста и опыта работы.

Методы и приемы в работе с детьми

- практические (игровые);
- экспериментирование;
- моделирование;
- воссоздание;
- преобразование; • конструирование.

Дидактическое обеспечение:

Теоретический материал:

- Справочники с теорией курса для учеников.

Дидактический материал:

- наглядные презентации, интерактивные задания к каждому уроку;
- макет для итогового сайта
- готовые опросы и проверочные задания для учеников с использованием онлайн-приложений
- материалы по аттестации.
- доступ к онлайн-кабинету с занятиями на сайте htmlacademy.ru для педагога и учеников с возможностью отслеживания прогресса учащихся;
- Методических указаний для учителя в текстовом виде, презентации с иллюстративным изложением теоретического материала.
- Упражнений на платформе «HTML Academy» с теоретическим и практическим материалом.
- Интерактивных проверочных заданий в приложениях Kahoot и Learning Apps.

4. Список литературы и интернет источников

Литература, рекомендованная для педагога (основная)

1. Бер, Бибо jQuery. Подробное руководство по продвинутому JavaScript / Бибо Бер. - М.: Символ-плюс, 2015. – 243 с.
2. Дакетт Джон HTML и CSS. Разработка и дизайн веб-сайтов. — Эксмо, 2017.
3. Климов, Александр JavaScript на примерах / Александр Климов. - М.: БХВПетербург, 2018. - 336 с.
4. Макфарланд Дэвид Сойер Новая большая книга CSS. — Питер, 2018

5. Вордерман Кэрол, Вудкок Джон, Макаманус Шон. Переводчик: Ломакин Станислав Программирование для детей Манн, Иванов и Фербер, 2015 г.
6. Голиков Д., Голиков А. Книга юных программистов на Scratch. Учебное пособие/Москва: Smashwords, 2014-128с.
7. Флэнаган Д. JavaScript. Подробное руководство. – СПб.: Символ+Плюс, 2013.
8. Фримен Э., Робсон Э. Изучаем программирование на JavaScript – СПб.: Питер, 2015.
9. Пахомова Н. Ю. Метод учебного проекта в образовательном учреждении: Пособие для учителей и студен-ов педагогических вузов. М.: Аркти, 2008. 112 с.

Литература для педагога (дополнительная)

1. Купер Нейт Как создать сайт. Комикс-путеводитель по HTML, CSS и WordPress. — Манн, Иванов, Фербер, 2019.
2. Программирование для детей. Перевод с английского Станислава Ломакина, Москва, «Манн, Иванов и Фербер», 2015 г.

Литература для учащихся и родителей (основная)

1. Морган Ник JavaScript для детей. Самоучитель по программированию. Манн, Иванов и Фербер, 2016. – 287 с.
2. Горячев А.В., Горина К.И., Волкова Т.О. Информатика («Информатика в играх и задачах»). – М.: Баласс, Школьный дом, 2010. – 64 с.
3. Филиппов С. А. Робототехника для детей и родителей. – 3-е изд. – СПб.: Наука, 2013.

Литература для учащихся и родителей (дополнительная)

1. Тетрадь с заданиями для развития детей. Игровая информатика. - в 2 частях. ИП Бурдина С.В.
2. Тур С.Н.: Учебник-тетрадь по информатике для 7 класса. / С.Н. Тур, Т.П. Бокучава. – СПб.: БХВ – Петербург, 2007. – 112 с.

Электронные ресурсы

1. <https://htmlacademy.ru/>
2. <http://htmlbook.ru/>
3. <https://web-standards.ru/>
4. <https://css-live.ru/>
5. <https://css-tricks.com/>
6. <https://alistapart.com/>
7. <https://www.smashingmagazine.com>

**Комплексы упражнений
для проведения физкультурминуток
для снятия напряжения рук, шеи, туловища и глаз**

Упражнения при утомлении

- Медленно опустить подбородок на грудь и оставаться в таком положении 5 сек. Прodelать 5 - 10 раз.
- Откинуться на спинку кресла, положить руки на бедра, закрыть глаза, расслабиться и посидеть так 10 - 15 сек.
- Выпрямить спину, теле расслабить, мягко прикрыть глаза. Медленно наклонять голову вперед, назад, вправо, влево.
- Сидя прямо с опущенными руками, резко напрячь мышцы всего тела. Затем быстро полностью расслабиться, опустить голову, закрыть глаза. Посидеть так 10 - 15 сек. Прodelать упражнение 2 - 4 раза.
- Сесть удобно, слегка расставив ноги. Руки положить на середину живота. Закрыть глаза и глубоко вздохнуть через нос. Задержать дыхание (насколько возможно). Медленно выдохнуть через рот (полностью). Прodelать упражнение 4 раза (если не возникнет головокружение).

Эффект: расслабление тела, снятие нервного напряжения, восстановление нормального ритма дыхания.

Упражнения для глаз

- Закрыть глаза, расслабить мышцы лба. медленно с напряжением сместить глазные яблоки в крайне левое положение, через 1 - 2 сек. так же перевести взгляд вправо. Прodelать 10 раз. Следить за тем, чтобы веки не подрагивали. Не щуриться.

Эффект: расслабление и укрепление глазных мышц, избавление от боли в глазах.

- Моргать в течении 1 - 2 мин. с напряжением закрывать на 3 - 5 сек. попеременно один и другой глаз.
- В течении 10 сек. несколько раз сильно зажмуриться.
- В течении 10 сек. менять направление взгляда: прямо, вправо, влево, вверх, вниз.
- Потереть ладони одну о другую, чтобы появилось ощущение тепла. Прикрыть ладонями глаза, скрестив пальцы в центре лба. Полностью исключить доступ света. На глаза и веки не нажимать. Расслабиться, дышать свободно. Побыть в таком положении 2 мин.

Эффект: химическое восстановление рецепторов глаз, расслабление глазных мышц, улучшение кровообращения в зрительно аппарате, избавление от ощущения усталости глаз.

Упражнения для головы и шеи

- Помассировать лицо, чтобы снять напряжение лицевых мышц.
- Надавливая пальцами на затылок в течении 10 сек. делать вращательные движения вправо, затем влево.

Эффект: расслабление мышц шее и лица.

- Закрыть глаза и сделать глубокий вдох. На выдохе медленно опустить подбородок, расслабить шею и плечи. Снова глубокий вдох, медленное круговое движение головой влево и выдох. Прodelать 3 раза влево, затем 3 раза вправо.

Эффект: расслабление мышц головы, шеи и плечевого пояса.

Упражнения для рук

- В положении сидя или стоя расположить руки перед лицом. Ладони наружу, пальцы выпрямлены. Напрячь ладони и запястья. Собрать пальцы в кулаки, быстро загибая их один за другим (начинать с мизинцев). Большие пальцы окажутся сверху. Сильно сжатые кулаки повернуть так, чтобы они "посмотрели" друг на друга. Движение - только в запястьях, локти не подвижны. Разжать кулаки, расслабить кисти. Прodelать упражнение еще несколько раз.

Эффект: снятие напряжения в кистях и запястьях.

- В положении сидя или стоя опустить руки вдоль тела. Расслабить их. Сделать глубокий вдох и на медленном выдохе в течение 10 - 15 сек слегка потрясти руками. Прodelать так несколько раз.

Эффект: избавление от усталости рук.

- Сцепить пальцы, соединить ладони и приподнять локти. Поворачивать кисти то пальцами внутрь (к груди), то наружу. Прodelать несколько раз, затем опустить руки и потрясти расслабленными кистями.

- Поцелкать пальцами обеих рук, перемещая большой палец поочередно на все другие пальцы.

- Широко расставить пальцы на напрячь кисти на 5 - 7 сек, затем сильно сжать пальцы в кулаки на 5 - 7 сек, после чего разжать кулаки и потрясти расслабленными кистями. Прodelать упражнение несколько раз.

Упражнения для туловища

- Встать прямо, слегка расставить ноги. Поднять руки вверх, подняться на носки и потянуться. Опустить, руки вдоль туловища, расслабиться. Прodelать 3 - 5 раз.

- Поднять плечи как можно выше и плавно отвести их назад, затем медленно выставить вперед. Прodelать 15 раз.

- Стоя нагнуться, приложить ладони к ногам позади колен. Втянуть живот и напрячь спину на 5 - 6 сек. Выпрямиться и расслабиться. Прodelать упражнение 3 - 5 раз.

- Встать прямо, ноги на ширине плеч. Развести руки в стороны на уровне плеч. Как можно больше повернуть туловище вправо, затем влево. Прodelать так 10 - 20 раз.

- Ноги на ширине плеч, слегка расслаблены и согнуты в коленях. Делая глубокий вдох, расслабиться. На выдохе поднять руки вверх, тянуть их к потолку. Ощутить напряжение в мышцах пальцев рук, плеч, спины и снова - глубокий вдох.

На выдохе наклониться вперед и коснуться руками пола перед носками туфель. Опустить голову, расслабиться. Вдох - и на выдохе выпрямиться. Прodelать упражнение 3 раза.

Эффект: расслабление мышц, распрямление позвоночника, улучшение кровообращения.

Кейс «Прогноз погоды по API»

О кейсе

Познакомиться с API, выяснить что они могут делать и как их использовать. Рассматриваем на примере API сайта OpenWeatherMap.org. Для получения доступа к сервису погоды придется пройти несложную процедуру регистрации на сайте OpenWeatherMap.org. Сформируем и отправим запрос, разберем ответный пакет в формате JSON, и получим текущую температуру с описанием состояния погоды.

Категория кейса

Углубленный, требуются базовая компьютерная грамотность, понимание основ HTML и CSS, основы JavaScript.

Возраст обучающихся – 12 - 17 лет

Место в структуре программы

Кейс выполняется как практическая работа после прохождения базовых тем по JavaScript, закрепление знаний по работе JS с массивами

Количество академических часов, на которые рассчитан кейс: от 6 до 8 час.

Учебно-тематическое планирование

Блок 1. Верстка страницы	
Предполагаемая продолжительность	Цель блока
90 - 180 мин	<i>Сделаем верстку страницы приложения "Погода".</i>
<p>Что делаем: <i>В зависимости от знаний HTML и CSS выполняем верстку страницы, будущего информера погоды. Сложность будет иметь 2 уровня – 1повторение верстки по шаблону, 2 верстка с разработка дополнительных полей, предложенных самостоятельно/</i></p>	
Блок 2. Взаимодействие JS, API	
Предполагаемая продолжительность	Цель блока
90-180 мин	<i>Научиться взаимодействовать с API по средством JS</i>
<p>Что делаем: <i>Пишем функционал для ранее созданной веб страницы, используя полученные базовые знания по JS. Разбираем примеры работы с API посредством JS. Сложность 2 уровня – 1 выполнение кода по предложенному шаблону, 2 пишем код с возможностью выбирать города в выпадающем списке - или предоставление пользователю возможности вводить имя города на странице, вывод дополнительной информации о погоде, снабдить страницу иконками, вывод страны, где находится город, сделать возможность получать прогноз погоды на несколько дней вперед.</i></p>	

Предполагаемые результаты:

Артефакты: Создание страницы сайта с информером погоды.

Soft skills:

Навыки самоорганизации

Умение работать с информацией

Креативные навыки

Критическое мышление

Hard skills:

Понимание основ работы с API

Умение взаимодействовать с API посредством JS

Навыки работы с JS

Дополнительно (вариативная часть)**Руководство наставника****Текст-легенда кейса**

Практика, практика и еще раз практика. Для успешного усвоения материала по JS необходим опыт практической работы, где ребенок самостоятельно проходит шаги по разработки своего, пусть и небольшого проекта. При написании клиентского JavaScript для веб-сайтов или приложений вы быстро столкнетесь с интерфейсами прикладного программирования (API). API-это программные средства для управления различными аспектами браузера и операционной системы, на которых работает сайт, или для управления данными с других веб-сайтов или служб. Разработка информера отличная возможность пополнить ваше портфолио законченными проектами, которые продемонстрируют ваш уровень вебразработчика.

Материалы в помощь:

Тема взаимодействия с API довольно хорошо освещена, ниже приведены примеры материала по данной тематике, вы можете подобрать по запросам более подходящую вам.

Клиентский веб API

https://developer.mozilla.org/ru/docs/Learn/JavaScript/Client-side_web_APIs

Информер погоды на сайте — api OpenWeatherMap

<https://active-vision.ru/blog/api-openweathermap/>

сервис OpenWeatherMap

<https://home.openweathermap.org/>

Обратить внимание:

Интеграция API OpenWeatherMap на свой сайт и создание полноценного информера будет проходить в 3 этапа.

1. Что такое API? Практика на примере OpenWeatherMap. Получение ключа OpenWeatherMap API. (Внимание ключ активируется не сразу, может пройти несколько часов)

2. Верстка страницы для информера. (В зависимости от уровня сложности будущего приложения, будет изменено количество элементов на странице)

3. Пишем программный код для получения, обработки и вывода данных на веб-страницу.

Руководство для обучающегося**Текст-легенда кейса**

Туристическая фирма предлагает разработать и разместить у себя на сайте информер погоды, показывающий различные данные по погоде в точках туристических маршрутов.

Жизненный цикл:

Проблематизация -> Целеполагание -> Поиск решения -> Планирование -> Реализация замысла -> Финализация

Кейс «Создание сайта-портфолио с помощью Flask»

О кейсе

С помощью данного кейса учащиеся создадут небольшой сайт-портфолио с использованием фреймворка Flask и SQLite в Python 3. На сайте будет размещена информация об учащихся. Примеры его работ. Подключена база данных для создания, редактирования и удаления постов.

Категория кейса

Углубленный, требуются базовая компьютерная грамотность, понимание основ HTML и CSS, основы программирования (понятия переменная, массив, функция, цикл, условия).

Примерный возраст обучающихся – 12 - 17 лет

Место в структуре программы

Кейс выполняется как практическая работа после прохождения базовых тем по HTML и CSS, закрепление знаний по работе с Python 3.

Количество академических часов, на которые рассчитан кейс: от 8 до 10 час.

Учебно-тематическое планирование

Блок 1. Введение. Flask.	
Предполагаемая продолжительность	Цель блока
45 мин.	Знакомство с веб-фреймворком Flask. Подготовка к работе.
<p>Что делаем: Говорим о Flask. что такое, для чего необходим. Установка и настройка необходимого программного обеспечения. Тест работоспособности.</p>	

Блок 2. Создание веб-приложения	
Предполагаемая продолжительность	Цель блока
135 - 180 мин	Создание базового приложения.
<p>Что делаем: На этом этапе создаем небольшое веб-приложение внутри файла Python. Тестируем его работу. Использование шаблонов HTML. Говорим о механизме шаблонов Jinja. Используем язык разметки CSS для оформления стиля приложения и приданию ему более привлекательного вида с помощью своего собственного дизайна. Используем инструментарием Bootstrap, который предлагает простые в использовании компоненты для оформления приложения.</p>	

Блок 3. База данных	
Предполагаемая продолжительность	Цель блока
135 - 180 мин	<i>Научиться работать с базой данных. Продемонстрировать настройку базы данных для хранения, наполнение базы, редактирование и удаление существующих постов.</i>
<p>Что делаем: Настраиваем базу данных для хранения данных, в случае с вашим приложением — это посты новостей. Наполняем базу несколькими примерами. Изменяем функцию просмотра <code>index()</code>, чтобы отображать все посты, внесенные в базу данных. Создаем новый маршрут Flask с функцией просмотра и новым шаблоном HTML для отображения отдельного поста блога по его <code>id</code>. Учимся быстро и удобно манипулировать значениями, что находятся в базе данных. Знакомимся с выборкой записей из базы данных. Добавляем дополнительный функционал к нашему веб-приложению. Возможность для редактирования и удаления определенных записей из базы данных.</p>	

Блок 4. Завершаем разработки и размещение сайта	
Предполагаемая продолжительность	Цель блока
45 мин.	<i>Учимся выгружать сайт на удаленный сервер Heroku</i>
<p>Что делаем: <i>Для публикации Flask приложения в Интернете нам потребуется - выгрузить сайт на хостинг. Будем использовать сервис Heroku. Тестируем результат.</i></p>	

Предполагаемые результаты обучающихся:

Артефакты: *личный сайт-портфолио с реализованной базой данных.*

Soft skills:

*Навыки самоорганизации
 Умение работать с информацией
 Креативные навыки
 Критическое мышление*

Hard skills:

*Знание основ и принципов программирования
 Опыт работы с HTML5, CSS3, JS, Python
 Знание фреймворка Bootstrap
 Знание фреймворка Flask
 Опыт работы с SQL
 Опыт верстки сайтов
 Работа с сервером Heroku*

Дополнительно (вариативная часть)

Руководство наставника

Текст-легенда кейса

Кейс погружает учащегося в мир веб разработки. Где ребенок окажется в роли веб разработчика и на практике познакомится с реальными задачами. Знания и умения работы с основными инструментами веб разработчика, знакомство с фреймворками, реализация собственного продукта дадут направление для дальнейшего развития.

Материалы в помощь:

Обзор кейса

Фреймворк Flask относится к так званым микрофреймворкам - минималистичные каркасы, которые обеспечивают лишь базовые возможности для реализации той или иной задачи. В стандартную поставку входит только самое необходимое, а все остальное при необходимости легко подключается. На Flask можно реализовать серьезный, масштабируемый проект. А для реализации таких небольших веб-приложений, как наш сайт, Flask подходит как нельзя лучше.

Примерные секции будущего сайта:

- Главная
- Резюме
- Портфолио
- Новости
- Контакты

В кейсе представлены важные концепции фреймворка Flask. Учащиеся научатся создавать веб приложение, запускать его на сервере разработки и на сервере Heroku. Научатся использовать механизм шаблонов Jinja для повторного использования файлов HTML. По итогам кейса учащиеся получают полностью работоспособный веб сайт, использующий базу данных SQLite для работы с новостями – создание, отображение, редактирование и удаление новостных постов.

Можно усложнить кейс, добавив аутентификацию, позволяющую разграничить права пользователей. Реализовать загрузку файлов пользователями.

Ссылки на дополнительную информацию:

[Документация Flask](#)

[Расширения Flask](#)

[Курс по языку Python](#)

[Программа PyCharm](#)

[Шаблонизатор Jinja](#)

[Уроки SQL для начинающих](#)

[Сервис Heroku](#)

[Heroku CLI](#)

[Установка Git](#)

Список литературы

1. Алексеев А. Введение в Web-дизайн. Учебное пособие. — М.: ДМК Пресс, 2019. — 184 с.
2. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. — М.: Эксмо, 2019. — 480 с.
3. Минник Крис, Титтел Эд. HTML5 и CSS3 для ч.айников. — М.: Диалектика, 2019. — 400 с.

4. Сырых Ю. А. Современный веб-дизайн. Настольный и мобильный. — М.: Диалектика, 2019. — 384 с
5. Flask Web Development Автор: Miguel Grinberg. Год издания: 2018. Язык: русский.
6. Flask: веб-разработка капля за каплей Год издания: 2016. Язык: русский.
7. Flask By Example Автор: Gareth Dwyer. Год издания: 2016. Язык: английский.

Шаги по реализации кейса:

Шаг 1 — Установка Flask

Шаг 2 — Создание базового приложения

Шаг 3 — Использование шаблонов HTML

Шаг 4 — Настройка базы данных

Шаг 5 — Отображение всех постов

Шаг 6 — Отображение отдельного поста

Шаг 7 — Изменение постов

Шаг 8 — Тестирование и публикация приложения в сети интернет

Руководство для обучающегося

Текст-легенда кейса

В большинстве общеобразовательных учреждений России, согласно нововведению Министерства образования и науки, создание портфолио для ученика является обязательным. По федеральному государственному образовательному стандарту оно является перспективной формой демонстрации индивидуальных достижений школьника и представляет собой набор документов, содержащих результаты учебной, творческой, спортивной, общественной жизни ребёнка. Как его организовать? Как презентовать? Где хранить? Эти вопросы решаются нашим кейсом «Создание сайта-портфолио с помощью Flask в Python 3».

Жизненный цикл:

Проблематизация -> Целеполагание -> Поиск решения -> Планирование -> Реализация замысла -> Финализация

Установка Flask

Для установки Flask используется команда:

```
pip install flask
```

Исходный код

Основной файл **app.py**

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return "Hello World"

@app.route('/about')
def about():
    return "About page"

if __name__ == "__main__":
```

```
app.run(debug=True)
```

Отслеживание URL

Пример отслеживания URL:

```
@app.route('/')
@app.route('/home')
def index():
    return render_template("index.html")
```

Шаблонизатор Jinja

Все шаблоны хранятся в одной общей папке *templates*.

В основном шаблоне можно прописать базовую структуру для всех HTML файлов:

```
<body>
{% block content %}
{% endblock %}
</body>
```

Внутри базового шаблона можно встроить HTML блоки и заменить содержимое *block content*. Чтобы это сделать создаем новый HTML файл и прописываем наследование структуры из базового файла:

```
{% extends "ПУТЬ К HTML ФАЙЛУ" %}

{% block content %}
<div>Здесь код, который будет добавлен</div>
{% endblock %}
```

Мы можем создавать небольшие кусочки кода и встраивать их в другие шаблоны. Для таких кусочков лучше создавать отдельную папку с названием *includes*. В папке можно создать простой HTML файл с небольшой Jinja разметкой:

```
{% block content %}
<div>Здесь код, который будет добавлен</div>
{% endblock %}
```

Для подключения таких файлов используйте директиву «include»:

```
{% include "ПУТЬ К ФАЙЛУ" %}
```

Работа с базой данных SQLite

Flask обеспечивает легкую и удобную работу со всеми популярными базами данных. Нам нужно подключить базу данных SQLite к нашему проекту, а также создать таблицу в базе данных и добавить в неё данные через пользовательский интерфейс.

Создание таблиц

Flask обладает хорошим функционалом по работе с базой данных и табличками в ней. Сперва необходимо подключиться к соответствующей базе данных и далее зарегистрировать её в проекте ч.е.р.е.з терминал. Далее в основном файле вы можете записать как один, так и несколько классов, каждый из которых будет выполнять роль определенной таблицы в базе данных.

Создание полей

Для описания полей внутри таблицы необходимо прописать переменные внутри классов. В каждую переменную устанавливается определенное значение, ч.т.о соответствует типу устанавливаемых данных в поле таблицы.

При получении объекта из базы данных всегда возвращается ID записи. С такой информацией особо ничего не сделать, поэтому лучше в классе модели дописывать магический метод «`__repr__`». В нём можно указать какое значение будет возвращаться при получении объекта из БД.

В методе ниже указано, ч.т.о будет возвращаться значение из поля «title»:

```
def __repr__(self):  
    return '<Article %r>' % self.id
```

Отображение данных из БД

Flask позволяет быстро и удобно манипулировать значениями, ч.т.о находятся в базе данных. За счёт SQLAlchemy можно выбирать, добавлять, удалять и редактировать данные всего за пару строк кода. В ходе реализации кейса мы познакомимся с выборкой записей из базы данных.

Необходимо использовать модели для взаимодействия с какой-либо таблицей в базе данных. Модель представляет из себя прямого проводника к определенной таблице в БД. ч.е.р.е.з модель вы можете создавать новые записи, обновлять уже существующие, удалять записи и получать записи для дальнейшего отображения в шаблонах.

Чтобы получить записи из таблицы, вам нужно создать объект на основе вашей модели и далее обратиться к одной из многих доступных функций:

```
model = User.query.first() # Первая запись  
model = User.query.all() # Все записи из таблицы
```

Чтобы передать данные в HTML шаблон, укажите второй параметр в функции `render_template()`:

```
model = User.query.all() # Создание объекта и получение всех записей
```

```
# Вызов шаблона и передача в него данных  
return render_template('html-template.html', parametr_name=model)
```

Работа с уже созданными записями проста и не требует особых усилий. ч.т.о.бы обновлять записи, вам изначально необходимо прописать нахождение нужной записи из БД и далее процесс обновления точно такой же, как и при добавлении новой записи.

```

@app.route('/posts/<int:id>/update', methods=['POST', 'GET'])
def post_update(id):
    article = Article.query.get(id)
    if request.method == "POST":
        article.title = request.form['title']

        try:
            db.session.commit()
            return redirect('/posts')
        except:
            return "При редактировании статьи произошла ошибка"

    else:
        return render_template("post_update.html", article=article)

```

Чтобы выполнить удаление данных, то вам также изначально необходимо получить определенную запись и далее выполнить её удаление:

```

# В функции принимаем ID определенной записи
@app.route('/posts/<int:id>/del')
def post_delete(id):
    article = Article.query.get_or_404(id)

    try:
        db.session.delete(article)
        db.session.commit()
        return redirect('/posts')
    except:
        return "При удалении статьи произошла ошибка"

```

Публикация на сервер

Для публикации Flask приложения в Интернете вам потребуется выполнить одно - выгрузить сайт на хостинг сервиса Heroku.

Публикация Flask проекта

Для начала в проект добавьте библиотеку gunicorn:

```
pip install gunicorn
```

Также в проекте необходимо создать файл «requirements.txt», а также файл «Procfile». Далее создайте приложение на Heroku и следуйте простым инструкциям по установке проекта на сервер данной системы:

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Clone the repository

Use Git to clone lesson-hard's source code to your local machine.

```
$ heroku git:clone -a lesson-hard (укажите название вашего проекта)
$ cd lesson-hard
```

Deploy your changes

Make some changes to the code you just cloned and deploy them to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Листинг файла app.py

```
from flask import Flask, render_template, url_for, request, redirect
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///blog.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

class Article(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(100), nullable=False)
    intro = db.Column(db.String(300), nullable=False)
    text = db.Column(db.Text, nullable=False)
    date = db.Column(db.DateTime, default=datetime.utcnow)

    def __repr__(self):
        return '<Article %r>' % self.id

@app.route('/')
@app.route('/index')
def index():
    return render_template("index.html")

@app.route('/about')
def about():
    return render_template("about.html")

@app.route('/modul')
def modul():
    return render_template("modul.html")

@app.route('/modul02')
def modul02():
```

```

        return render_template("modul02.html")

@app.route('/modul03')
def modul03():
    return render_template("modul03.html")

@app.route('/posts')
def posts():
    articles = Article.query.order_by(Article.date.desc()).all()
    return render_template("posts.html", articles=articles)

@app.route('/posts/<int:id>')
def posts_detail(id):
    article = Article.query.get(id)
    return render_template("post_detail.html", article=article)

@app.route('/posts/<int:id>/del')
def post_delete(id):
    article = Article.query.get_or_404(id)

    try:
        db.session.delete(article)
        db.session.commit()
        return redirect('/posts')
    except:
        return "При удалении новости произошла ошибка"

@app.route('/posts/<int:id>/update', methods=['POST', 'GET'])
def post_update(id):
    article = Article.query.get(id)
    if request.method == "POST":
        article.title = request.form['title']
        article.intro = request.form['intro']
        article.text = request.form['text']

        try:
            db.session.commit()
            return redirect('/posts')
        except:
            return "При редактировании новости, произошла оштбка"

    else:
        return render_template("post_update.html", article=article)

@app.route('/foto')
def foto():
    return render_template("foto.html")

@app.route('/user/<string:name>/<int:id>')
def user(name, id):
    return "User page:" + name + " - " + str(id)

@app.route('/article', methods=['POST', 'GET'])
def article():
    if request.method == "POST":
        title = request.form['title']
        intro = request.form['intro']
        text = request.form['text']

```

```

    article = Article(title=title, intro=intro, text=text)

    try:
        db.session.add(article)
        db.session.commit()
        return redirect('/posts')
    except:
        return "Ошибка"

else:
    return render_template("article.html")

if __name__ == "__main__":
    app.run(debug=True)

```

Пример шаблона html страницы

```

<!doctype html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <!-- CSS only -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
crossorigin="anonymous">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/main.css')
}}">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
<div class="container">
<div class="d-flex flex-column flex-md-row align-items-center pb-3 mb-4 border-
bottom">
    <a href="/" class="d-flex align-items-center text-dark text-decoration-
none">
        <span class="fs-4">IT kvant</span>
    </a>

    <nav class="d-inline-flex mt-2 mt-md-0 ms-md-auto">

        <a class="me-3 py-2 text-dark text-decoration-none"
href="/index">Информация</a>
        <a class="me-3 py-2 text-dark text-decoration-none nav-link dropdown-
toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-
toggle="dropdown" aria-expanded="false">
            Модули
        </a>
        <ul class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
            <li><a class="dropdown-item" href="/modul">Вводный</a></li>
            <li><a class="dropdown-item" href="/modul02">Углубленный</a></li>
            <li><a class="dropdown-item" href="/modul03">Проектный</a></li>
        </ul>
        <a class="me-3 py-2 text-dark text-decoration-none"
href="/foto">Фотогалерея</a>

```

```

        <a class="me-3 py-2 text-dark text-decoration-none"
href="/about">Контакты</a>
        <a class="me-3 py-2 text-dark text-decoration-none"
href="/posts">Новости</a>
        <a class="btn btn-warning" href="/article">Создать новость</a>
    </nav>
</div>

{% block body %}{% endblock %}
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
pprn3073KE6t16bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
</body>
</html>

```

Блок новостей

```

{% extends 'base.html' %}

{% block title %}
Новости
{% endblock %}

{% block body %}

<div class="container mt-5">
<h1>Новости</h1>
    {% if articles|length >0 %}
        {% for el in articles %}
            <div class="alert alert-info">
                <h2>{{el.title}}</h2>
                <p>{{ el.intro}}</p>
                <p>{{ el.date.date()}}</p>
                <a href="/posts/{{el.id}}" class="btn btn-warning">Подробнее</a>

            </div>
        {% endfor %}
        {% else %}
            <p>У нас еще нет новостей на сайте</p>
        {% endif %}
    </div>
{% endblock %}

```

Файл подробной новости

```

{% extends 'base.html' %}

{% block title %}
{{ article.title }}
{% endblock %}

{% block body %}

<div class="container mt-5">
<h1>{{ article.title }}</h1>

```

```

    <div class="alert alert-info">
      <p>{{ article.text }}</p>
      <p>{{ article.date.date() }}</p>
      <a href="/posts/{{article.id}}/del" class="btn btn-danger">Удалить</a>
      <a href="/posts/{{article.id}}/update" class="btn btn-
success">Редактировать</a>

    </div>

</div>
{% endblock %}

```

Файл редактирования новости

```

{% extends 'base.html' %}

{% block title %}
о нас
{% endblock %}

{% block body %}
<h1>Регистрация</h1>
<form method="post">
  <input type="text" name="title" id="title" class="form-control"
placeholder="Название"><br>
  <textarea name="intro" id="intro" class="form-control"
placeholder="Анонс"></textarea><br>
  <textarea name="text" name="title" id="text" class="form-control"
placeholder="Основной текст"></textarea><br>
  <input type="submit" class="btn btn-success" value="Отправить">
</form>

{% endblock %}</html>

```

Содержание файла Procfile

```
web: unicorn app:app
```

Пример содержания файла requirements.txt

```

aiogram==2.21
aiohttp==3.8.1
aiosignal==1.2.0
async-timeout==4.0.2
attrs==21.4.0
Babel==2.9.1
certifi==2022.6.15
charset-normalizer==2.1.0
click==8.1.3
colorama==0.4.5
Flask==2.1.2
Flask-SQLAlchemy==2.5.1
frozenlist==1.3.0
greenlet==1.1.2
unicorn==20.1.0
idna==3.3
importlib-metadata==4.12.0
itsdangerous==2.1.2
Jinja2==3.1.2

```

```
MarkupSafe==2.1.1
multidict==6.0.2
pyTelegramBotAPI==4.6.0
pytz==2022.1
requests==2.28.1
SQLAlchemy==1.4.39
urllib3==1.26.10
Werkzeug==2.1.2
yarl==1.7.2
zipp==3.8.0
```