

Муниципальное автономное учреждение дополнительного образования
«Полазненский центр дополнительного образования детей
«Школа технического резерва»

Принята на заседании
педагогического совета

от «13» сентября 2017 г.
Протокол № 4

УТВЕРЖДАЮ:

Директор МАУДО «ПЦДОД «ШТР»

 Э.Г. Гонтарь

«13» сентября 2017 г.



**Дополнительная общеобразовательная
общеразвивающая программа технической
направленности
«WEB-программирование»**

Возраст обучающихся: 14 – 17 лет

Срок реализации: 1 год

Автор-составитель:
Бажина Наталья Николаевна
педагог дополнительного
образования

Полазна
2017

Оглавление

Введение.....	3
1. Пояснительная записка.....	4
1.1 Актуальность, новизна и необходимость разработки программы	4
1.2 Цель и задачи программы.....	4
2. Организация образовательного процесса	4
2.1 Учет особенностей контингента	4
2.2 Особенности реализации программы	5
2.3 Критерии оценки эффективности реализации программы	5
3. Учебно-методический план.....	5
4. Календарно-тематическое планирование	6
5. Содержание программы.....	8
Список литературы	11

Введение

Мир переходит в цифровую эру и некоторые люди уже живут в Интернете больше, чем в «реальной жизни». И мы видим, как бизнес реагирует на эти изменения:

- все магазины постепенно переезжают в Сеть или обзаводятся онлайн-представительством;
- маркетологи всё чаще отдают предпочтение рекламным сетям в интернете, оставляя без внимания привычные радио и телевидение;
- некоторые компании рождаются в сети развиваясь в огромные порталы принося своим владельцам многомиллиардные прибыли.

Такое развитие веб-пространства порождает закономерное желание «влиться» в этот процесс и научиться создавать сайты самостоятельно.

Веб-программирование – раздел программирования, ориентированный на разработку веб-приложений (программ, обеспечивающих функционирование динамических сайтов Всемирной паутины).

Курс предназначен для новичков в области веб-программирования, будет полезен как для школьников старших классов, так и для всех людей, которые интересуются веб-технологиями или планируют начать карьеру в этой индустрии. Полученный уровень подготовки позволит учащимся в дальнейшем с профориентацией.

HTML - это не язык программирования, а язык разметки. С его помощью нельзя сделать полноценный сайт, можно только создать «оболочку», где элементы взаимодействия с пользователем будут видны, но не будут работать. Тем не менее, веб-программирование невозможно без знания HTML.

Таким образом, курс рассчитан не только на изучение основ программирования на HTML, но и оформление веб-страниц с помощью CSS, а также создание интерактивных элементов с помощью JavaScript.

1. Пояснительная записка

1.1 Актуальность, новизна и необходимость разработки программы

В наше время стремительного прогресса просто красиво оформленный текст и картинки на веб-сайте уже никого не удивят. Да и требования к сайтам, надо сказать, очень изменились - и теперь для успешного представительства компании в Интернет необходимо предоставить своим посетителям различные возможности: обратную связь, подписки, рассылки, поиск по сайту и многое другое.

При помощи обычного html этого не достичь, ведь html - это всего лишь язык разметки - инструмент для создания гиперссылок, вставки изображений, таблиц и др. Это хорошо видно на примере. С помощью языка html легко и быстро можно сделать форму для отправки какого-либо запроса. Такая форма будет иметь все необходимые атрибуты: и поле для ввода текста, и кнопку отправки. Однако при нажатии на такую кнопку в большинстве случаев не произойдет ровным счетом ничего - ведь не был задан сценарий действий, которые следует выполнить, чтобы получить результат. Поэтому создание интерактивных компонентов - это задача уже для веб-программирования.

1.2 Цель и задачи программы

Цель образовательной программы - научить "с нуля" создавать современные веб-интерфейсы, работая с кодом вручную.

Задачи образовательной программы:

- Зажечь в детях тягу к сфере информационных технологий, на примере педагогов образовательного учреждения, показать перспективы занятий программированием, его применимость в современном обществе. Заложить основу и направление будущего инженерного развития.

- Предоставить возможность для изучения программирования. Заложить принципы самостоятельного обучения.

- Дети смогут самостоятельно создавать веб-страницы начального и среднего уровня сложности.

- Привить культуру командной работы.

2. Организация образовательного процесса

Занятия проходят один раз в неделю по два академических часа. Программа реализуется в течение учебного года.

2.1 Учет особенностей контингента

Программа рассчитана на возраст детей от 14 до 17 лет.

Данная программа одинаково подходит как юношам, так и девушкам. Среди необходимых профессиональных качеств следует выделить:

- Усидчивость. Большая часть работы проходит в монотонных процессах за компьютером.

- Ответственность. Взявшись за проект, программист должен довести его до конца и предоставить высококлассный результат, даже если работа ему не слишком интересна.

- Хорошая память. Существует масса языков программирования, при этом один специалист должен знать минимум 3-7 самых востребованных.

- Креативность и новаторский подход.

- Высокий уровень внимания. Один неправильный символ, и придется переписывать сайт с нуля.
- Умение работать с большими объемами информации. Хороший веб-программист ведет сразу несколько проектов.
- Аналитический склад ума. Это одно из основных качеств, помогающих быстро писать сайты и принимать правильные решения.

2.2 Особенности реализации программы

Специфика предмета позволяет проводить занятия в различной форме. Могут быть использованы интерактивные игры, требующие технического творчества, соревнования для развития лидерских и командных навыков, научно-практические семинары могут быть полезны для привития ответственности и опыта выступления перед публикой. Данная программа позволяет использовать все перечисленные подходы в обучении детей.

Также следует учитывать, что прогресс не стоит на месте, особенно это касается информационных технологий. Поэтому педагог обязан постоянно следить за инновациями в области информационных технологий и стараться максимально внедрить новые подходы во время реализации программы. Так, например, обновление программного обеспечения позволит использовать самые передовые методики в сфере информационных технологий.

2.3 Критерии оценки эффективности реализации программы

Критерии оценки эффективности реализации программы. Каждое занятие состоит из изучения нового материала, работы над созданием программы, где ученики применяют полученные знания и небольших соревнований в конце урока. Такое построение урока дает понимание ценности потраченного времени и прививает ответственность за результат. Большие проекты также проверяются в форме конференций, но на уровне школы. Поэтому рекомендуется как можно чаще участвовать в городских, краевых и российских соревнованиях. Проводить собственные мероприятия и форумы с привлечением экспертов, специалистов, прессы и высоких чинов нашего государственного аппарата. Чем больше важности будет оказано деятельности учеников, тем выше результаты будут получены в реализации программы. При подведении итогов следует отмечать удовлетворенность учениками от прохождения образовательной программы, активность и инициативность в образовательном процессе, желание публично демонстрировать результат.

3. Учебно-методический план

№	Тема	Всего часов, из них		Форма контроля
		Теория	Практика	
1	Среда и стандарты Интернет-разработок.	15	40	контрольные задания
2	Стилевое и скриптовое воздействие на документ.	25	56	контрольные задания
	Итого	40	96	

4. Календарно-тематическое планирование

№	Тема	Содержание	Всего часов, из них		Форма контроля
			Теория	Практика	
1	Среда и стандарты Интернет-разработок.	1.1 Ключевая терминология сферы Интернет-проектов.	15	40	контрольные задания
		1.2 Принципы верстки страницы.			
		1.3 Шапка HTML страницы (<HEAD>).			
		1.4 Узнаваемость HTML документа.			
		1.5 Тело HTML страницы (<BODY>).			
		1.6 Ключевые теги разметки (<BODY>).			
		1.7 Структурное форматирование текста.			
		1.8 Маркеры элементов списка.			
		1.9 Внедрение иллюстраций тегом .			
		1.10 Публикация парных изображений.			
		1.11 Табличная структура.			
2	Стилевое и скриптовое воздействие на документ.	2.1 Необходимость стилизации контента.	25	56	контрольные задания
		2.2 Режимы подключения стилей.			
		2.3 Правила записи команд стилизации.			
		2.4 К чему применяется стилевое форматирование.			
		2.5 Стилиевые классы (работают всегда).			
		2.6 Стилиевые идентификаторы (для блочной верстки).			
		2.7 Важнейшие единицы измерения в CSS.			
		2.8 Способы описания цветов в CSS.			

	2.9 Команды стилевого форматирования.			
	2.10 CSS управление текстурой.			
	2.11 Стилизация поведения ссылок.			
	2.12 Обработчики и скриптовое управление.			
	2.13 Технологии реализации скриптовых внедрений в HTML.			
	2.14 Возможное расположение скриптов в структуре страницы.			
	2.15 Синтаксис команд внедрения.			
	2.16 Способы назначения обработчиков событий.			
	2.17 Общие правила языка JavaScript.			
	2.18 Направленность команд языка JavaScript.			
	2.19 Прямое воздействие на свойства объекта.			
	2.20 Воздействие на свойства объекта через функцию.			
	2.21 Скриптовые библиотеки.			
	Итого		40	96

5. Содержание программы

Тема №1 «Среда и стандарты Интернет-разработок»

1.1 Ключевая терминология сферы Интернет-проектов.

Основные понятия, такие как: WEB-дизайн, Интернет, Сайт, Интранет, Интранет-сайт, Гипертекст, HTML, Браузер, Протокол, WWW, Сервер, Домен, IP-адрес, Публикация сайта.

1.2 Принципы верстки страницы.

Расширение «html» и «htm». Знакомство с тегами.

1.3 Шапка HTML страницы (<HEAD>).

Направление текста. Назначение языка и кодировки. Кэширование документа. Переадресация на другой ресурс. Программа генератор и авторские права.

1.4 Узнаваемость HTML документа.

Заголовок страницы. Внедрение команд стилизации. Внедрение исполняемого скриптового контента.

1.5 Тело HTML страницы (<BODY>).

Внешний вид по умолчанию. Цветовое решение // атрибуты. Отступы от границ // атрибуты.

1.6 Ключевые теги разметки (<BODY>).

Физическое форматирование. Манипуляции с габаритами шрифта. Специальное цитирование текста. Универсальное форматирование.

1.7 Структурное форматирование текста.

Заголовки и подзаголовки. Текстовые абзацы (параграфы). Универсальный контейнер (блок). Контейнер текстовых фрагментов.

1.8 Маркеры элементов списка.

OL – нумерованный список (маркер цифровой или буквенный).

UL – маркированный список (маркер графический или примитив).

1.9 Внедрение иллюстраций тегом .

Основные атрибуты.

1.10 Публикация парных изображений.

Названия файлов иллюстраций. Публикация превьюшки в контейнере ссылки. Бордюры на навигационных изображениях.

1.11 Табличная структура.

table - главный контейнер табличных данных,

caption - подпись (надпись) таблицы,

tr, td, th - общие параметры контейнеров (строка, ячейка тела и шапки),

td, th - дополнительные параметры ячеек тела и шапки.

Тема №2 «Стилевое и скриптовое воздействие на документ»

2.1 Необходимость стилизации контента.

Версии развития технологии CSS.

2.2 Режимы подключения стилей.

Подключение атрибутом style в конкретном теге. Размещение стиля в шапке HTML страницы. Подключение внешнего стиля (рекомендуется).

2.3 Правила записи команд стилизации.

Правила стилизации не чувствительны к регистру. Возможна запись в одну строку (в конкретном теге). Для стиливых контейнеров и файлов рекомендуется построчная запись с табуляцией.

2.4 К чему применяется стиливое форматирование.

Ко всей области документа. К любому структурному тегу страницы (в атрибуте style). К структурным тегам («для всех тегов Р» - реагируют все абзацы). К элементам в определенном состоянии («ссылки А, на которые наезжает курсор»). К вложенным

элементам («ссылки А внутри абзацев Р»). К «дочерним» элементам («первый в списке»). К фрагментам тегов («первая строка»).

2.5 *Стилевые классы.*

Объявление стиливых классов. Примеры подключения классов.

2.6 *Стилевые идентификаторы (для блочной верстки).*

Объявление уникальных стиливых идентификаторов (для блоков). Подключение стиливого идентификатора.

2.7 *Важнейшие единицы измерения в CSS.*

Единицы измерения размерности.

2.8 *Способы описания цветов в CSS.*

Разбор возможных вариантов.

2.9 *Команды стиливого форматирования.*

«Цвет и фон» важнейшие стиливые параметры.

«Шрифт» важнейшие стиливые параметры.

«Бордюр» важнейшие стиливые параметры.

«Контейнеры» важнейшие стиливые параметры.

2.10 *CSS управление текстурой.*

Режимы позиционирования текстуры background-position.

Направление и ограниченное background-repeat.

Оригинальное сочетание фона и текстуры (пример, "загнутый уголок").

2.11 *Стилизация поведения ссылок.*

A:link - все ссылки документа (по умолчанию)

A:visited - посещенные ссылки (там уже были и вернулись)

A:active - ссылка в момент щелчка мышью

A:hover - ссылка, на которую наехал курсор

A:focus - выбранная ссылка до потери фокуса (навигация табуляцией)

Дополнительная стилизация групп ссылок.

2.12 *Обработчики и скриптовое управление.*

Строгость синтаксиса и чувствительность языка к регистру. Проблемы вложенности строковых выражений (кавычек).

2.13 *Технологии реализации скриптовых внедрений в HTML.*

JavaScript - межплатформенный стандарт браузеров (начиная с версии 1.2).

VBScript - технология Microsoft (Инtranет).

JScript и LiveScript - устаревшие технологии (заменены JavaScript).

SSI - Внедрения в серверные документы (подстановка контента на сервере).

2.14 *Возможное расположение скриптов в структуре страницы.*

В шапке - рекомендуется для автономных сценариев.

В теле - эффективно для простейших действий и вызовов функций.

Во внешнем файле - универсальные сценарии и популярные библиотеки.

2.15 *Синтаксис команд внедрения.*

```
<p onClick="код">...</p>
```

```
<img onMouseOver="код1" onMouseOut="код2">
```

```
<script language="JavaScript">...</script>
```

```
<script language="JavaScript" src="mylib.js">...</script>
```

2.16 *Способы назначения обработчиков событий.*

Через прямое воздействие. Через внешний указатель (идентификатор) / прямое указание. Через внешний указатель (идентификатор) / известная функция.

2.17 *Общие правила языка JavaScript.*

Операторы - учитывают регистр. Разделитель – точка с запятой ";". Наглядность кода: простые конструкции – однострочная запись, сложные конструкции - вложенность выделяется табуляцией.

2.18 Направленность команд языка JavaScript.

А) Визуальные эффекты (замена, появление, смещение, подсказки):

- собственные разработки,
- подключаемые библиотеки.

Б) Навигационные эффекты:

- интерактивное меню.

В) Сметные расчеты:

- собственный калькулятор.

Г) Динамическая подгрузка контента без смены адреса:

- технологии семейства JQUERY + AJAX.

Д) Проверка корректности заполнения форм:

- простейшие правила (валидаторы + заполненность),
- сверка с базой данных (серверный сценарий).

Е) Построение фрагментов страницы:

- табличные и регулярные структуры.

Ж) Определение вида и версии броузера:

- для переадресации (сложный контент).

Е) Рекламные возможности:

- ротация рекламных элементов (появление, замена),
- мониторинг поведения клиентов.

2.19 Прямое воздействие на свойства объекта.

Пример: замена изображений при наведении курсора мыши.

2.20 Воздействие на свойства объекта через функцию.

Пример: подсветка информационных строк таблицы.

2.21 Скриптовые библиотеки.

Наиболее известна библиотека конструирования:

- <http://jquery.com/>

Часто используются и прочие специализированные библиотеки эффектов:

- <http://visuallightbox.com/lightbox-mac-style-demo.html>
- <http://highslide.com>

Список литературы

1. Кошин В.А. Интернет-программирование - Пермь, 2016
2. <http://edunews.ru/professii/obzor/tehnicheskie/web-programmist.html>
3. <https://openedu.ru/course/ITMOUniversity/WEBDEV/>
4. <https://progstudy.ru/index.php/sm/article/veb-programmirovanie-uroki-html>
5. <http://webstudio2u.net/ru/programming/127-programming.html>