

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«УЧЕБНЫЙ ЦЕНТР СКБ КОНТУР»



Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»

Т.В. Рубан

1 сентября 2023 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

дополнительная общеразвивающая программа

FRONTEND-РАЗРАБОТКА

Документ: Свидетельство

Форма обучения: заочная с использованием дистанционных образовательных технологий, электронного обучения

Срок обучения: 3 недели

Объем программы: 32 ак. ч.

Москва, 2023 г.

Оглавление

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	3
1.1. Назначение программы.....	3
1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	3
2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы	3
2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы	3
2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы	4
2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	4
3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА.....	5
3.1. Учебный план.....	5
3.2. Содержание	6
3.3 Календарный учебный график.....	11
4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ	12
4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса	12
4.2. Требования к материально-техническим условиям	12
4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	14
4.4. Общие требования к организации образовательного процесса	15
5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	16
5.1. Формы контроля знаний и требования к его проведению	16
5.2. Критерии оценки знаний слушателей	16
5.3. Фонд оценочных средств	17

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Назначение программы

Дополнительная общеобразовательная программа дополнительная общеразвивающая программа «Frontend-разработка» направлена на получение практических навыков и теоретических знаний, необходимых для тех, кто хочет познакомиться с инструментами веб-программирования HTML, CSS, JavaScript и освоить навыки разработки и тестирования веб-приложений.

Программа регламентирует цели, планируемые результаты, содержание, условия и технологии реализации образовательного процесса, оценку качества подготовки слушателей и включает в себя: учебный план, фонды оценочных средств, программу итоговой аттестации, календарный учебный график и другие материалы, обеспечивающие качество подготовки слушателей.

1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

1. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ.
2. Федеральный закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
3. Приказ Минпросвещения РФ от 27.07.2022 №629 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам".

2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Основная цель программы – дать слушателям базовые знания о разработке и тестировании web-приложений с помощью HTML, CSS, JavaScript.

Задачи дисциплины

В процессе изучения курса «Основы HTML и CSS» необходимо решить следующие задачи и рассказать слушателям:

- о работе с HTML, CSS на базовом уровне;
- о базовой структуре и основных элементах HTML-документа;
- о разработке web-приложений;
- о способах применения CSS-стилей к html-элементам;
- об использовании JavaScript при разработке и тестировании веб-приложений.

2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

К освоению дополнительной общеобразовательной программы – дополнительной общеразвивающей программы допускаются любые лица без предъявления требований к уровню образования.

Возраст слушателей: 18 лет и старше.

2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Нормативный срок освоения программы – 3 недели.

2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Объем образовательной программы составляет 32 академических часа. Из них 20 часов – самостоятельная работа, 12 академических часов – работа на образовательной онлайн-платформе.

2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.

В результате обучения слушатели должны будут овладеть следующими знаниями, умениями и навыками:

Знать:

- основы работы с HTML, CSS, JavaScript;
- основы разработки веб-приложений.

Уметь:

- создавать HTML-документы;
- применять CSS для стилизации веб-страниц.

Владеть:

- навыками разработки web-приложений;
- навыками использования JavaScript и CSS при тестировании веб-приложений.

3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

3.1. Учебный план

для заочной формы обучения с ДОТ и ЭО

№ п/п	Наименование разделов и дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
1.	HTML/CSS	18	12	6	Зачет
1.1	Работа с DevTools	3	2	1	Тестирование
1.2	Основы HTML	3	2	1	Тестирование
1.3	Изображения и ссылки в HTML	3	2	1	Тестирование
1.4	Формы и таблицы в HTML	3	2	1	Тестирование
1.5	Синтаксис и способы подключения CSS	3	2	1	Тестирование
1.6	Селекторы	3	2	1	Тестирование
2.	JavaScript	12	8	4	Зачет
2.1	Введение в JavaScript	3	2	1	Тестирование
2.2	Типы данных	3	2	1	Тестирование
2.3	Объекты: основы	3	2	1	Тестирование
2.4	Обработка ошибок	3	2	1	Тестирование
	ИТОГОВАЯ АТТЕСТАЦИЯ	2	–	2	Зачет
–	Всего:	32	20	12	–

3.2. Содержание

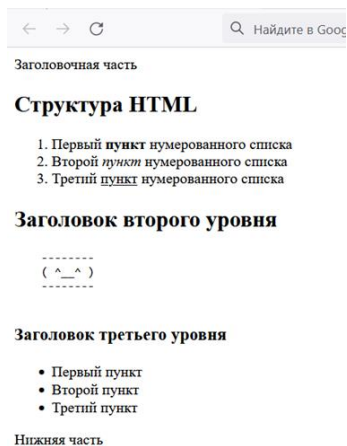
Тема 1. «HTML/CSS»

Урок 1.1. Работа с DevTools

- Инструменты для разработки и отладки кода: Chrome DevTools, Firefox DevTools.
- Использование Chrome DevTools, Firefox DevTools.
- Консоль разработчика. Использование консоли разработчика для тестирования и поиска ошибок в браузере.
- Практическое задание:
 - 1) Откройте консоль разработчика и вызовите через консоль все виды сообщений: `console.error`, `console.warn`, `console.debug`, `console.info`.
 - 2) Создайте HTML-файл с js-кодом внутри, используя тег `script`, впишите туда все команды урока, откройте файл в браузере и посмотрите результат вывода в консоли разработчика.
 - 3) На любом сайте просмотрите все вкладки в инструментах разработчика, стараясь извлечь максимум полезной информации о веб-приложении. Кратко написать, что полезного вы узнали о сайте при помощи инструментов разработчика.

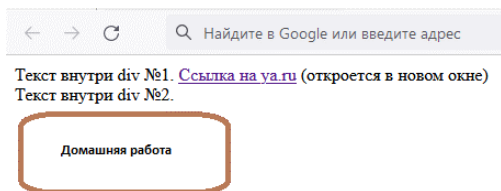
Урок 1.2. Основы HTML

- Назначение и появление HTML.
- Возможности HTML.
- Базовая структура и основные элементы HTML-документа.
- Основные HTML-теги. Теги для работы с текстом. Теги для работы со списками.
- Практическое задание: создать HTML-документ по макету. Для выполнения задания используйте материалы урока. Результат пришлите в виде HTML-кода.



Урок 1.3. Изображения и ссылки в HTML

- Что такое гиперссылка, виды гиперссылок.
- Гиперссылки в HTML. Способ открытия ссылки.
- Работа с изображениями в HTML. Теги и способы встраивания изображений.
- Метатеги, комментарии и фреймы.
- Практическое задание. Сверстайте HTML-страницу по макету



Для выполнения задания используйте материалы урока. Результат пришлите в виде HTML-кода.

Урок 1.4. Формы и таблицы в HTML

- Формы в HTML. Клиентская часть формы.
- Установка и запуск сервера.
- Серверная часть формы.
- Таблицы в HTML. Теги таблицы, атрибуты тегов таблицы.
- Практическая работа:

1) Реализуйте форму обратной связи «Анкета знакомств». Прототип:

Проверьте, что форма отправляет письма на локальный почтовый сервер. Для выполнения задания используйте материалы урока. Результат опубликуйте на github'е и пришлите ссылку на репозиторий.

2) Сверстайте таблицу по прототипу:

Музыкальный альбом	Название трека	Исполнители
	Начало альбома	
	Тучи	Иванушки
	Беги	Иванушки

Результат пришлите в виде HTML-кода. Результат опубликуйте на github'е и пришлите ссылку на репозиторий.

Урок 1.5. Синтаксис и способы подключения CSS

- Язык таблицы стилей CSS (Cascading Style Sheets). Виды селекторов.
- Структура CSS-файла.
- Основные категории и свойства CSS.
- Способы встраивания стилей CSS в html-страницу.

— Порядок применения стилей.

— Практическое задание:

1) В первом параграфе мы создавали HTML-страницу.

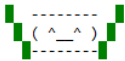
Раскройте ее согласно прототипу (используя полученные знания про CSS):

Верхняя часть

Структура HTML

1. Первый пункт нумерованного списка
2. Второй пункт нумерованного списка
3. Третий пункт нумерованного списка

Заголовок второго уровня



Заголовок третьего уровня

- Первый пункт
- Второй пункт
- Третий пункт

Нижняя часть

Результат пришлите в виде текста (сначала HTML-код, затем CSS-стили).

2) Найдите и перечислите все возможные значения-типы атрибута rel (помимо stylesheet). Опишите назначение каждого типа.

3) Найдите ошибки в этом HTML-файле и исправьте их:

<HTML>

<head>

<style>

```
html{color:#FF0;}
```

```
body {
```

```
color: purple; }
```

```
p { color: black;
```

```
}
```

```
.superClass {
```

```
text-color: #FF0;
```

```
}
```

</style>

</head>

<body>

```
<div style="font-size:20px;color:grey"><span style="font-size:20px;color:red;">Какой-<span  
class="superClass">то</span> текст</span div></div>
```

```
</bOdy>
```

</html>

Результат пришлите в виде исправленного текста.

Урок 1.6. Селекторы

- Селекторы
 - CSS-селекторы
 - XPath-селекторы
 - Функции в XPath-селекторах
 - Способы генерации селекторов.
 - Практическое задание:
 - 1) Найдите и составьте список самых популярных псевдоклассов (псевдоэлементов). Результат пришлите в виде текста.
 - 2) Составьте выражения для поиска элементов с текстом «параграф» двумя способами: при помощи CSS и при помощи XPath.


```
<div id="myId" class="homework">
  <div class="point">Параграф первый.</div>
  <div class="point">Параграф второй.</div>
  <div class="point">Параграф третий.</div>
</div>
```
- Результат пришлите в виде текста.

Тема 2. «JavaScript»

Урок 2.1. Введение в JavaScript

- Введение в JavaScript.
- Применение JavaScript и CSS при тестировании веб-приложений.
- Синтаксис JavaScript.
- Базовые конструкции (оператор, условие, цикл, функции).
- Логирование в JavaScript.
- Практическое задание:
 - 1) Реализуйте функцию сложения двух чисел. Если хотя бы одно из чисел отрицательное, функция должна возвращать сообщение «Отрицательные числа складывать не буду». Результат можно проверить в тренажере.
 - 2) Реализуйте функцию, в которой есть цикл от 1 до 6. Нужно создать массив и положить в него только четные числа. В результате функция должна возвращать массив из четных чисел: 2 4 6. Результат можно проверить в тренажере.
 - 3) Откройте консоль разработчика и вызовите через консоль все виды сообщений: console.error, console.warn, console.debug, console.info. Результат пришлите в виде скриншота.

Урок 2.2. Типы данных

- Типы данных в JavaScript
- Преобразование типов данных
- Операции с базовыми типами данных
- Практическое задание:
 - 1) Петя плохо выучил JavaScript, написал программу и забыл проверить. Найдите ошибки в программе Пети и исправьте их.
 - 2) Написать свою программу с использованием типов данных Boolean, Number, String. Результат пришлите в виде текста.

Урок 2.3 Объекты: основы

- Класс, свойства и методы.
- Наследование классов.
- Приватные методы и свойства.
- Статические методы и свойства.
- Практическое задание:
 1. Создайте класс Student со свойствами: имя, оценка. Создайте пару студентов и выведите значения (имена и их оценки) в консоль.
 2. В созданном классе Student создайте метод, который выводит в консоль «Молодец!», если оценка у студента «5».

Урок 2.4. Обработка ошибок

- Ошибки в JavaScript.
- Перехват и обработка ошибок.
- Генерация ошибок.
- Ошибки в HTML.
- Ошибки в CSS.
- Практическое задание:
 - 1) Напишите код, вызывающий все типы встроенных ошибок (SyntaxError, ReferenceError, TypeError, RangeError, URIError), рассмотренные в уроке. В консоли разработчика должны быть вызваны все типы ошибок.
 - 2) Напишите код, который генерирует ошибки всех типов встроенных ошибок при помощи оператора throw.
 - 3) Итоговое задание. На основе полученных в курсе знаний сверстайте веб-страницу по прототипу.



Результат пришлите в виде текста (если текстовых файлов несколько, скопируйте тексты в один).

3.3 Календарный учебный график

Календарный график обучения является примерным, составляется и утверждается для каждой группы.

Срок освоения программы — 3 недели. Начало обучения — по мере набора группы. Примерный режим занятий: не более 8 академических часов в день, до 16 часов в неделю. Промежуточная и итоговые аттестации проводятся согласно графику.

Примерный график освоения программы:

№	Темы / дни	ВР	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1.	HTML/CSS	РП	2	1	1	1	1																
		СР						2	2	2	2	2	2										
2.	JavaScript	РП												2	1	1							
		СР															2	1	1	2	1	1	
3.	Итоговая аттестация	РП																					2

4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса

Требования к образованию и обучению лица, занимающего должность преподавателя: высшее образование — специалитет или магистратура, направленность (профиль) которого, как правило, соответствует преподаваемому учебному курсу, дисциплине (модулю).

Дополнительное профессиональное образование — профессиональная переподготовка, направленность (профиль) которой соответствует преподаваемому учебному курсу, дисциплине (модулю).

Педагогические работники обязаны проходить в установленном законодательством Российской Федерации порядке обучение и проверку знаний и навыков в области охраны труда.

Рекомендуется обучение по дополнительным профессиональным программам по профилю педагогической деятельности не реже чем один раз в три года.

Требования к опыту практической работы: при несоответствии направленности (профиля) образования преподаваемому учебному курсу, дисциплине (модулю) — опыт работы в области профессиональной деятельности, осваиваемой слушателями или соответствующей преподаваемому учебному курсу, дисциплине (модулю).

Преподаватель: стаж работы в образовательной организации не менее одного года; при наличии ученой степени (звания) — без предъявления требований к стажу работы.

Особые условия допуска к работе: отсутствие ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации

Прохождение обязательных предварительных (при поступлении на работу) и периодических медицинских осмотров (обследований), а также внеочередных медицинских осмотров (обследований) в порядке, установленном законодательством Российской Федерации

Прохождение в установленном законодательством Российской Федерации порядке аттестации на соответствие занимаемой должности.

4.2. Требования к материально-техническим условиям

Организация проводит занятия по адресу: г. Москва, ул. Суцеский Вал, д. 18. Аудитории для занятий расположены на 11-м этаже здания.

Все занимаемые помещения соответствуют обязательным нормам пожарной безопасности и требованиям санитарно-эпидемиологических служб. Помещения имеют централизованные системы водоснабжения, отопления и канализации. Воздухообмен помещений обеспечивается современными системами кондиционирования, за счет приточно-вытяжной вентиляционной системы.

Учебным центром СКБ Контур заключен договор с организацией общественного питания о возможности обеспечения слушателей питанием.

В учебной аудитории проводятся лекции и практические занятия. Аудитория оснащена столами и стульями, в составе учебного оснащения маркерная доска и флипчарт, в случае необходимости подключается мультимедийный проектор, слушателям предоставляются компьютеры.

Компьютерная сеть учебного центра оснащена необходимым оборудованием для доступа в интернет по выделенному каналу. На каждом компьютере обеспечен постоянный доступ к компьютерной программе «Контур.Школа».

Для проведения вебинаров и онлайн-трансляций используется оснащенная современным оборудованием видеостудия:

- помещение оборудовано посадочными местами для спикера(ов);
- спикеру предоставляется персональный компьютер с соответствующими мультимедийными характеристиками (Intel Core i3 либо идентичные по характеристикам, оперативная память: от 4 Гб и выше для всех ОС), со стабильным соединением с сетью Интернет на скорости не менее 1 Мбит/с;
- видеочасть (максимальное разрешение видео — не менее 3840 x 2160).

Размещение материалов вебинаров и доступ к ним участников обеспечивает техническая платформа (сайт, система управления сайтом, другие технические средства):

1. Трансляция вебинара в режиме реального времени.
2. Хранение, систематизация записей вебинаров, с предоставлением участникам возможности просмотра записи онлайн.
3. Хранение, систематизация и доступ к скачиванию материалов учебных программ.
4. Напоминание участникам о предстоящем вебинаре за 1 час до начала мероприятия.
5. Использование защищенных соединений, передача и прием видео и звука по протоколам RTMP(S) или аналогичным.
6. Управление качеством и разрешением передаваемого/принимаемого видео вплоть до разрешения HD 720p на каждого участника мероприятия (адаптивный стриминг).
7. Обмен короткими текстовыми сообщениями (чат).
8. Осуществление записи мероприятий в формате, не требующем конвертации для проигрывания (mp4, AVI, WMA и т.д.).
9. Система регистрации на вебинар.
10. Техническое сопровождение проведения вебинара.
11. Отображение числа участников.
12. Техническая доступность услуги не менее 99,8% времени.
13. Устойчивость при проведении вебинара при одновременном подключении до 3000 участников.
14. Возможность участия пользователей на вебинарах в браузерах Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari с установленным плагином Adobe Flash Player.
15. Передача аудио- и видеосообщения на персональные компьютеры участников реализована при скорости интернет-соединения не менее 134 кбит/с.

Основные функции программы Контур.Школа:

1. Размещение расписания и описания учебных программ и условий обучения.
2. Онлайн-трансляция учебных занятий с возможностью обратной связи.
3. Размещение тестов и проведение онлайн-тестирования.
4. Размещение и выбор образовательного контента и заданий для слушателей.
5. Хранение учебно-методических материалов.
6. Обратная связь слушателей к организаторам и преподавателям.
7. Автоматическая фиксация хода учебного процесса, промежуточных и итоговых результатов слушателей.
8. Хранение информации о ходе учебного процесса и результатов обучения в течение периода обучения.
9. Сбор и хранение заявок на обучение и сведений о слушателях.
10. Создание и актуализация контента и учебно-методических материалов.
11. Информационно-консультационное обслуживание слушателей.

4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Дополнительная общеобразовательная программа – дополнительная общеразвивающая программа обеспечивается учебно-методическими материалами по всем модулям образовательной программы.

Фонд учебно-научной библиотеки содержит основную и дополнительную учебную, учебно-методическую, научную литературу, справочно-библиографические и периодические издания (в том числе и на электронных носителях) по всем темам и дисциплинам реализуемой программы.

Нормативно-правовая база

1. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 №149-ФЗ
2. ГОСТ Р ИСО 9241-1 S I - 2014 «ЭРГОНОМИКА ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕК — СИСТЕМА», часть 151 «Руководство по проектированию пользовательских интерфейсов сети Интернет»
3. ГОСТ Р ИСО/МЭК 23026-2015 «Интернет-ресурсы. Основные требования к дизайну и контенту»

Список литературы

1. JavaScript с нуля до профи / [Роб Персиваль]; СПб.: Питер; 2023г. – 480с.
2. JavaScript с нуля/ [Кирупа Чиннатхамби]; СПб.: Питер, 2021. — 400 с.
3. JavaScript для профессиональных веб-разработчиков. 4-е международное издание/ [Мэтт Фрисби]; СПб.: Питер, 2022. — 1168 с.
4. JavaScript. Визуальные редакторы: учебное пособие для СПО / [В. В. Янцев]; Санкт-Петербург: Лань, 2022. — 168 с.
5. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков Учебное пособие / [Кириченко А.В.]; СПб.: Наука и техника; 2021 г. – 432 с.
6. Лучший учебник html. Основы html/ Электронный ресурс.
https://mathsciences.at.ua/luchshij_uchebnik-2_html.pdf
7. Справочник HTML. Кратко, быстро, под рукой / [Кириченко А.В., Дубовик Е.В.]; СПб.: Наука и техника; 2021 г. - 288 с.

8. Справочник JavaScript. Кратко, быстро, под рукой / [Никольский А.П., Дубовик Е. В.]; СПб.: Наука и Техника; 2021. - 304 с.
9. Учебник HTML для начинающих/ [Семикопенко Алексей Алексеевич]. Электронный ресурс. <https://doskol.narod.ru/FILES/HTML.pdf>

Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий», №2, 2023г. <http://www.vkit.ru/index.php/archive-rus/1228-02-february>
2. Научно-практический журнал «Программные продукты и системы» №1, 2023г. <http://www.swsys.ru/index.php>

Интернет-ресурсы

1. HTML и работа с таблицами <https://otus.ru/journal/html-i-rabota-s-tablicami/>
2. JavaScript и условные операторы <https://otus.ru/journal/javascript-i-uslovyne-operator/>
3. Основы программирования на HTML <https://otus.ru/journal/osnovy-programmirovaniya-na-html/>
4. Особенности и методы Object в Java и JavaScript <https://otus.ru/journal/osobennosti-i-metody-object-v-java-i-javascript/>
5. Учебник HTML для начинающих <https://msiter.ru/tutorials/html-nachalnogo-urovnya>

4.4. Общие требования к организации образовательного процесса

При реализации дополнительной общеобразовательной программы – дополнительной общеразвивающей программы используется лекционно-семинарская система, практические занятия, индивидуальные и групповые консультации, мастер-классы и другие интерактивные формы обучения. Интегративно-дифференцированная организация занятий в процессе подготовки слушателей позволяет не только адаптировать образовательный процесс к индивидуальным особенностям и интересам слушателей, но и обеспечить свободу выбора и вариативность образования, сформировать у слушателей стремление к самообразованию, способствовать решению индивидуальных задач развития личности.

Особую значимость на уровне технологии обучения имеет организационно-педагогическое условие, предполагающее организацию самостоятельной работы слушателей как средство формирования профессиональных компетенций. Самостоятельная работа слушателей является обязательной составляющей образовательного процесса подготовки слушателей дополнительной общеобразовательной программы – дополнительной общеразвивающей программы.

Применение дистанционных образовательных технологий и электронного обучения

Дистанционные образовательные технологии применяются частично.

В учебном процессе могут использоваться следующие организационные формы учебной деятельности:

- обзорные (установочные) лекции с использованием дистанционных образовательных технологий;
- самостоятельная работа с СДО: работа с электронным учебником, просмотр видеолекций и др.;
- самостоятельная работа с программами контроля знаний (тестами);
- консультации (индивидуальные с применением электронных средств, групповые и предэкзаменационные);
- семинары с использованием дистанционных образовательных технологий;
- коллоквиумы;

- итоговые аттестационные работы.

5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

5.1. Формы контроля знаний и требования к его проведению

Для проведения промежуточной и итоговой аттестации программы разработан фонд оценочных средств по программе, являющийся неотъемлемой частью учебно-методического комплекса.

Объектами оценивания выступают:

- степень освоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы, активность на занятиях.

Текущий контроль знаний слушателей проводится преподавателем, ведущим занятия в учебной группе, на протяжении всего обучения по программе.

Текущий контроль знаний включает в себя наблюдение преподавателя за учебной работой слушателей и проверку качества знаний, умений и навыков, которыми они овладели на определенном этапе обучения посредством выполнения упражнений на практических занятиях и в иных формах, установленных преподавателем.

Промежуточная аттестация — оценка качества усвоения слушателями содержания учебных блоков непосредственно по завершении их освоения, проводимая в форме зачета посредством тестирования или в иных формах, в соответствии с учебным планом и учебно-тематическим планом.

Итоговая аттестация — процедура, проводимая с целью установления уровня знаний, слушателей с учетом прогнозируемых результатов обучения и требований к результатам освоения образовательной программы. Итоговая аттестация слушателей осуществляется в форме зачета посредством тестирования.

Слушатель допускается к итоговой аттестации после изучения тем образовательной программы в объеме, предусмотренном для лекционных и практических занятий.

Слушателям, освоившим дополнительную общеобразовательную программу - дополнительную общеразвивающую программу по теме «Frontend-разработка», и успешно прошедшим итоговую аттестацию, выдается свидетельство установленного образца с указанием названия программы, календарного периода обучения, длительности обучения в академических часах.

Для аттестации слушателей на соответствие их персональных достижений требованиям соответствующей ОП созданы фонды оценочных средств, включающие типовые задания, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций.

Фонды оценочных средств соответствуют целям и задачам программы подготовки специалиста, учебному плану и обеспечивают оценку качества общепрофессиональных и профессиональных компетенций, приобретаемых слушателями.

5.2. Критерии оценки знаний слушателей

Предмет оценивания (компетенции)	Объект оценивания (навыки)	Показатель оценки (знания, умения)
Работа на языках программирования HTML, CSS, JavaScript	Применение знаний по основам языков программирования для выполнения разных задач при разработке и тестировании веб-приложений	<p>Знания:</p> <ul style="list-style-type: none"> • Основы языков программирования HTML и CSS, JavaScript. • Базовые концепции веб-программирования, их применение при разработке и тестировании веб-приложений. <p>Умения:</p> <ul style="list-style-type: none"> • Разрабатывать веб-приложения на языках программирования HTML, CSS, JavaScript. • Читать код, созданный на языке программирования HTML, CSS, JavaScript. • Соблюдать минимальные стандарты оформления кода. • Тестировать приложения с использованием языков веб-программирования.

Критерии оценки аттестации – зачета:

1. Оценка «**Зачтено**» выставляется слушателю, продемонстрировавшему твердое и всесторонние знания материалы, умение применять полученные в рамках занятий практические навыки и умения. Достижения за период обучения и результаты текущей аттестации демонстрировали отличный уровень знаний и умений слушателей. Не менее 80% правильных ответов при решении тестов.

2. Оценка «**Не зачтено**» выставляется слушателю, который в недостаточной мере овладел теоретическим материалом по дисциплине, допустил ряд грубых ошибок при выполнении практических заданий, а также не выполнил требований, предъявляемых к промежуточной аттестации. Достижения за период обучения и результаты текущей аттестации демонстрировали неудовлетворительный уровень знаний и умений слушателя. Менее 80% правильных ответов при решении тестов.

5.3. Фонд оценочных средств

Тест к уроку 1.1. «Работа с DevTools»

1. Как открыть консоль разработчика?
 1. Нажать Ctrl + Shift + I
 2. Нажать Ctrl + Shift + L
 3. Нажать F2
 4. Нажать на меню-троеточие и выбрать «Вспомогательные инструменты»
2. Какие вкладки не скрыты по умолчанию?
 1. Регистратор, Статистика производительности
 2. Быстрый исходный код, Поиск
 3. Анимации, Покрытие
 4. Сеть, Источники

3. Где найти справку по сочетаниям клавиш?
 1. Правый верхний угол — кнопка-троеточие — Дополнительные настройки — Сочетания клавиш
 2. Нажать F1
 3. Правый верхний угол — меню-троеточие — Сочетания клавиш
 4. Меню-троеточие — Настройки — Сочетания клавиш
4. Для чего не предназначен Chrome DevTools?
 1. Chrome DevTools используется для тестирования и отладки кода
 2. Инструменты разработчика помогают находить ошибки
 3. DevTools открывается клавишей F12
 4. DevTools используется для автоматического написания кода
5. Что нельзя делать с сочетаниями клавиш?
 1. Добавлять
 2. Удалять
 3. Изменять
 4. Изменять порядок
6. Какие виды сообщений можно увидеть в консоли разработчика?
 1. Debug, Error, Message
 2. Fatal, Error, Log
 3. Message, Info, Log
 4. Error, Warning, Info

Тест к уроку 1.2. «Основы HTML»

1. Какие возможности имеет HTML?
 1. Таблицы
 2. Гиперссылки
 3. Объекты
 4. Все перечисленные
2. Что такое DOM?
 1. HTML-документ
 2. HTML-модель
 3. Describe Open Move
 4. Document object model
3. Какие теги не могут содержать в себе footer?
 1. Header
 2. Footer
 3. Header, footer
 4. Head
4. Какими тегами можно сделать полужирное начертание шрифта?
 1. U, bold
 2. Strong, em
 3. Em, i
 4. Strong, b
5. В каких тегах игнорируются переносы строк и пробелы?

1. Pre, sub
2. Code, p
3. Br, pre
4. Code, pre

Тест к уроку 1.3. «Изображения и ссылки в HTML»

1. Каких ссылок не бывает?
 1. Якорных
 2. Внешних
 3. Анкорных
 4. Блочных
2. Какое значение атрибута target нужно использовать, чтобы страница открывалась в новом окне?
 1. parent
 2. _self
 3. blank
 4. _blank
3. Выберите способы встраивания изображений.
 1. Img, insert, map
 2. Media, picture, img
 3. Img, picture, map
 4. Css, img, source
4. Какой атрибут использовать для задания кодировки?
 1. Meta
 2. Utf-8
 3. Frameset
 4. Charset
5. Какие символы используются для вставки комментария?
 1. <> ... <>
 2. // ... //
 3. <!-- .. -->
 4. <!-- ... -->

Тест к уроку 1.4. «Формы и таблицы в HTML»

1. Какой тег чаще всего используется внутри form?
 1. Option
 2. Select
 3. Fieldset
 4. Input
2. Какой атрибут позволяет в обработчике получить значение поля?
 1. id
 2. class
 3. type
 4. name

3. Тег `fieldset` нужен для того, чтобы...
 1. Установить набор форм
 2. Определить форму
 3. Разграничить элементы формы
 4. Определить содержание формы
4. Какой атрибут есть у `table`, `tr` и `td`?
 1. `Abbr`
 2. `Char`
 3. `Background`
 4. `Align`
5. Для чего нужен тег `th`?
 1. Это лишний тег
 2. Для определения ячейки
 3. Для задания таблицы
 4. Для задания заголовка таблицы

Тест к уроку 1.5. «Синтаксис и способы подключения CSS»

1. Какого вида селектор используется в этом коде?

```
span {color: blue;}
```

 1. Селектор элемента
 2. Селектор класса
 3. Селектор идентификатора
 4. Универсальный селектор
2. Что такое атрибут?
 1. Устаревший элемент, который уже не используется
 2. Стил для экрана монитора
 3. То же самое, что тег
 4. То, что указывается внутри скобок тега
3. Какие значения не могут быть у атрибута `media`?
 1. `all`, `tv`, `braille`
 2. `speech`, `print`, `screen`
 3. `tty`, `tv`, `tti`
 4. `projection`, `handheld`, `screen`
4. Для чего нужен CSS?
 1. Чтобы разметить HTML-теги на странице
 2. Для установки правил для обработки исключений
 3. Для стилизации C#-компонентов
 4. Чтобы задать цвет, расположение элементов
5. Какого атрибута не существует у тега `style`?
 1. `type`
 2. `media`
 3. Всех перечисленных
 4. `mime`

Тест к уроку 1.6. «Селекторы»

1. Какого вида селектора не существует?
 1. Math
 2. XPath
 3. CSS
 4. Универсальный
2. Что такое XPath?
 1. Express Path Language
 2. X Path Language
 3. XML-разметка документа
 4. Язык запросов к элементам страницы
3. Какой XPath не найдет элемент на странице?
<div class="test">

<p>Параграф единственный и неповторимый!</p>

</div>
 1. /html/body/div/p
 2. //div/p
 3. body/div/p
 4. //body/*/p
4. Что не умеет XPath, но умеет CSS?
 1. Искать по селектору идентификатора
 2. Искать по селектору класса
 3. Искать по универсальному селектору
 4. Искать по селектору псевдокласса
5. Псевдоклассы придуманы для того, чтобы...
 1. Применять к элементам стили, которые не входят в обычные CSS-стандарты
 2. Применять стили к элементам, находящимся на одном уровне вложенности
 3. Применять стили ко всем элементам на странице сразу
 4. Применять стили к элементам, находящимся в определенном состоянии или порядке

Тест к теме 1. «HTML/CSS»

1. Как открыть консоль разработчика?
 1. Нажать Ctrl + Shift + I
 2. Нажать Ctrl + Shift + L
 3. Нажать F2
 4. Нажать на меню-троеточие и выбрать «Вспомогательные инструменты»
2. Какое утверждение является ошибочным?
 1. Chrome DevTools используется для тестирования и отладки кода
 2. Инструменты разработчика помогают находить ошибки
 3. DevTools открывается клавишей F12

4. DevTools используется для автоматического написания кода
3. Что такое DOM?
 1. HTML-документ
 2. HTML-модель
 3. Describe Open Move
 4. Document object model
4. Какими тегами можно сделать полужирное начертание шрифта?
 1. U, bold
 2. Strong, em
 3. Em, i
 4. Strong, b
5. Каких ссылок не бывает?
 1. Якорных
 2. Внешних
 3. Анкорных
 4. Блочных
6. Какие символы используются для вставки комментария?
 1. `<> ... <>`
 2. `// ... //`
 3. `<!-- .. -->`
 4. `<!-- ... -->`
7. Тег fieldset нужен для того, чтобы...
 1. Установить набор форм
 2. Определить форму
 3. Разграничить элементы формы
 4. Определить содержание формы
8. Для чего нужен тег th?
 1. Это лишний тег
 2. Для определения ячейки
 3. Для задания таблицы
 4. Для задания заголовка таблицы
9. Что такое атрибут?
 1. Устаревший элемент, который уже не используется
 2. Стил для экрана монитора
 3. То же самое, что тег
 4. То, что указывается внутри скобок тега
10. Для чего нужен CSS?
 1. Чтобы разметить HTML-теги на странице
 2. Для установки правил для обработки исключений
 3. Для стилизации C#-компонентов
 4. Чтобы задать цвет, расположение элементов
11. Какого вида селектора не существует?
 1. Math
 2. XPath

3. CSS
 4. Универсальный
12. Что не умеет XPath, но умеет CSS?
1. Искать по селектору идентификатора
 2. Искать по селектору класса
 3. Искать по универсальному селектору
 4. Искать по селектору псевдокласса

Тест к уроку 2.1. «Введение в JavaScript»

1. Какие ключевые слова существуют в JavaScript?
 1. function, if, typeof
 2. javascript, js, language
 3. importer, exporter, how
 4. start, finish, continue
2. Какие виды сообщений можно увидеть в консоли разработчика?
 1. Debug, Error, Message
 2. Fatal, Error, Log
 3. Message, Info, Log
 4. Error, Warning, Info
3. Как называется объект для доступа к средствам отладки?
 1. Debug
 2. Log
 3. Console
 4. Error
4. Какие стили программирования поддерживает JavaScript?
 1. JavaScript поддерживает объектное, обобщенно-компонентное, объектное и классическое программирование
 2. JavaScript поддерживает объектно-ориентированное, компонентно-ориентированное и обобщенное программирование (тут неверное — компонентно-ориентированное)
 3. JavaScript поддерживает обобщенно-компонентное, объектное и компонентное программирование
 4. JavaScript поддерживает объектно-ориентированное, функциональное и событийно-ориентированное программирование
5. Как вызвать вид сообщения «Ошибка» в консоли браузера?
 1. `var error = "error";
console.log(error);`
 2. `console.log(new Error);`
 3. `console.log("error");`
 4. `console.error("error");`
6. Когда код выполнится без ошибок?
 1. `const test;
console.log(test);`
 2. `var test = 'test';
console.log(tost);`
 3. `const test = 'test';
console.log(test);`
 4. `var test;
console.log(test);`

Тест к уроку 2.2. «Типы данных»

1. Что будет выведено в консоли?

```
var v1 = Symbol("value2");
```

```
var v2 = Symbol("value2");
```

```
var r = v1 == v2;
```

```
console.log(r);
```

1. false
 2. true
 3. null
 4. Будет ошибка
2. Для чего нужны типы данных в JavaScript?
1. Чтобы разграничить диапазоны возможных значений и операций
 2. Чтобы продемонстрировать структуры данных
 3. Для отладки и тестирования программы
 4. Для обозначения операций и операндов
3. Что будет выведено в консоли?
- ```
var v = true;
v = false;
v + true;
console.log(v);
```
1. 0
  2. true
  3. false
  4. 1
4. Каких типов данных не существует в JavaScript?
1. Boolean, Number, String
  2. Null, Object, BigInt
  3. Undefined, Symbol, BigInt
  4. Byte, Long, Char
5. В каком случае используется тип данных BigInt?
1. var bi = 5;
  2. var x = 5.05;
  3. let y = 5B;
  4. var bi = 5n;

### Тест к уроку 2.3 «Объекты: основы»

1. Для чего нужны методы класса?

1. Для совершения действий над объектом
2. Для хранения информации об объекте
3. Для создания экземпляра класса
4. Для вызова действия внутри другого класса



2. Что будет выведено в консоль?
- ```
const car = class MyCar {  
  model = "Volvo";  
}  
var car1 = new MyCar();  
console.log(car1.model);
```
1. Ничего
 2. 'MyCar' has already been declared
 3. Volvo
 4. MyCar is not defined
3. Выберите корректную(-ые) часть(-и) кода.
1. `var my = new My("value");`
 2. `var myValue = new My("key", "value");`
 3. Все корректные
 4. `this.something = something;`
4. Что такое класс?
1. Класс — это цикл
 2. Класс — это генератор
 3. Класс — это класс!
 4. Класс — это функция
5. Когда код выполнится без ошибок?
1.

```
class M{constructor(n){this.n=n;}}  
console.log(k.n);
```
 2.

```
class M{constructor(n){this.n=n;}}  
console.log(new m("n"));
```
 3.

```
class M{constructor(nn){this.nn=nn;}  
var s = new M("test it!");  
  
console.log(s.nn);
```
 4.

```
class M{constructor(n){this.n=n;}}  
var k = new M("n");  
console.log(k.n);
```

Тест к уроку 2.4. «Обработка ошибок»

1. Какого типа ошибок не существует в JavaScript?
 1. `ExceptionError`
 2. `TypeError`
 3. `URIError`
 4. `ReferenceError`
2. Что будет выведено в консоль?

```
decodeURIComponent('%test%');
```

 1. Ничего
 2. `%test%`

3. test
4. Uncaught URIError: URI malformed
3. Что будет выведено в консоль?

```
const myError = new Error('Общая пользовательская ошибка!');
console.log(myError.name);
```

 1. Общая пользовательская ошибка!
 2. Ошибка
 3. Error
 4. Ничего
4. Какие бывают направления ошибок в JS?
 1. Встроенные и системные
 2. Пользовательские и системные
 3. Встроенные и логические
 4. Пользовательские и встроенные
5. Какой код вызовет ошибку SyntaxError?
 1.

```
var opечатka = ['1', '2']
delete opечатka[0];
```
 2.

```
var abrakadabra = ['1', '2'];
delete arr[0];
```
 3.

```
var arr = ['1', '2'];
delete arr[0];
```
 4.

```
var arr = ['1', '2'];
delite arr[0];
```

Тест к теме 2. «JavaScript»

1. Какие виды сообщений можно увидеть в консоли разработчика?
 1. Debug, Error, Message
 2. Fatal, Error, Log
 3. Message, Info, Log
 4. Error, Warning, Info
2. Какие стили программирования поддерживает JavaScript?
 1. JavaScript поддерживает объектное, обобщенно-компонентное, объектное и классическое программирование
 2. JavaScript поддерживает объектно-ориентированное, компонентно-ориентированное и обобщенное программирование (тут неверное — компонентно-ориентированное)
 3. JavaScript поддерживает обобщенно-компонентное, объектное и компонентное программирование
 4. JavaScript поддерживает объектно-ориентированное, функциональное и событийно-ориентированное программирование
3. Что будет выведено в консоли?

```
var v1 = Symbol("value2");

var v2 = Symbol("value2");
```

```
var r = v1 == v2;
```

```
console.log(r);
```

1. false
 2. true
 3. null
 4. Будет ошибка
4. Каких типов данных не существует в JavaScript?
1. Boolean, Number, String
 2. Null, Object, BigInt
 3. Undefined, Symbol, BigInt
 4. Byte, Long, Char
5. Для чего нужны методы класса?
1. Для совершения действий над объектом
 2. Для хранения информации об объекте
 3. Для создания экземпляра класса
 4. Для вызова действия внутри другого класса
6. Выберите корректную(-ые) часть(-и) кода.
1. `var my = new My("value");`
 2. `var myValue = new My("key", "value");`
 3. Все корректные
 4. `this.something = something;`
7. Какого типа ошибок не существует в JavaScript?
1. `ExceptionError`
 2. `TypeError`
 3. `URIError`
 4. `ReferenceError`
8. Что будет выведено в консоль?
- ```
const myError = new Error('Общая пользовательская ошибка!');

console.log(myError.name);
```
1. Общая пользовательская ошибка!
  2. Ошибка
  3. `Error`
  4. Ничего

### Итоговое тестирование

1. Какие вкладки не скрыты по умолчанию?
  1. Регистратор, Статистика производительности
  2. Быстрый исходный код, Поиск
  3. Анимации, Покрытие
  4. Сеть, Источники
2. Что нельзя делать с сочетаниями клавиш?

1. Добавлять
  2. Удалять
  3. Изменять
  4. Изменять порядок
3. Какие возможности имеет HTML?
1. Таблицы
  2. Гиперссылки
  3. Объекты
  4. Все перечисленные
4. В каких тегах игнорируются переносы строк и пробелы?
1. Pre, sub
  2. Code, p
  3. Br, pre
  4. Code, pre
5. Какое значение атрибута target нужно использовать, чтобы страница открывалась в новом окне?
1. parent
  2. \_self
  3. blank
  4. \_blank
6. Какой атрибут использовать для задания кодировки?
1. Meta
  2. Utf-8
  3. Frameset
  4. Charset
7. Какой атрибут позволяет в обработчике получить значение поля?
1. id
  2. class
  3. type
  4. name
8. Какой атрибут есть у table, tr и td?
1. Abbr
  2. Char
  3. Background
  4. Align
9. Какого вида селектор используется в этом коде?  
`span {color: blue;}`
1. Селектор элемента
  2. Селектор класса
  3. Селектор идентификатора
  4. Универсальный селектор
10. Какие значения не могут быть у атрибута media?
1. all, tv, braille
  2. speech, print, screen

3. tty, tv, tti
  4. projection, handheld, screen
11. Что такое XPath?
1. Express Path Language
  2. X Path Language
  3. XML-разметка документа
  4. Язык запросов к элементам страницы
12. Как называется объект для доступа к средствам отладки?
1. Debug
  2. Log
  3. Console
  4. Error
13. Что будет выведено в консоли?
- ```
var v = true;
v = false;
v + true;
console.log(v);
```
1. 0
 2. true
 3. false
 4. 1
14. Что будет выведено в консоль?
- ```
const car = class MyCar {
 model = "Volvo";
}
var car1 = new MyCar();
console.log(car1.model);
```
1. Ничего
  2. 'MyCar' has already been declared
  3. Volvo
  4. MyCar is not defined
15. Какие бывают направления ошибок в JS?
1. Встроенные и системные
  2. Пользовательские и системные
  3. Встроенные и логические
  4. Пользовательские и встроенные