

Негосударственное образовательное частное учреждение высшего образования  
«Московский финансово-промышленный университет «Синергия»

**Дополнительная общеобразовательная программа  
«Разработка интерактивного сайта на JavaScript»**

Москва, 2022

## **ОГЛАВЛЕНИЕ**

<b>ПОЯСНИТЕЛЬНАЯ ЗАПИСКА</b>	<b>3</b>
<b>УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН</b>	<b>7</b>
<b>КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК</b>	<b>8</b>
<b>РАБОЧАЯ ПРОГРАММА</b>	<b>9</b>
<b>ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ</b>	<b>13</b>
<b>УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ</b>	<b>14</b>
<b>ФОРМЫ АТТЕСТАЦИИ</b>	<b>16</b>
<b>ОЦЕНОЧНЫЕ МАТЕРИАЛЫ, ВКЛЮЧАЯ ПРИМЕРЫ КОНТРОЛЬНЫХ ЗАДАНИЙ</b>	<b>17</b>
<b>ПЕРЕЧЕНЬ ИСТОЧНИКОВ ИНФОРМАЦИОННОГО СОПРОВОЖДЕНИЯ</b>	<b>21</b>
<b>КОМПЕТЕНЦИИ, ФОРМИРУЕМЫЕ В ПРОЦЕССЕ ОБУЧЕНИЯ</b>	<b>22</b>

# **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

## **Аннотация**

Программа «Разработка интерактивного сайта на JavaScript» предназначена для учащихся 8-11 классов. Этот курс будет полезен тем, кто интересуется областью разработки и поддержки веб-приложений.

Курс не требует специальной дополнительной подготовки, достаточно знаний предметной области информатики на базовом уровне.

Обучение на курсе будет проходить в формате вебинаров, поэтому для успешного прохождения курса потребуется наличие компьютера и высокоскоростного доступа в интернет. После окончания занятия запись вебинара будет доступна на платформе.

В ходе изучения учебных материалов программы слушатели освоят навыки разработки программного кода используя синтаксис JavaScript, научатся работать с системой контроля версий Git, фреймворками Node.JS, Vue.JS, React, а также познакомятся с версткой сайтов при помощи HTML и CSS. После прохождения программы слушатель сможет писать элементарные веб-приложения на языке JavaScript, применять фреймворки своих проектов, а также собирать полноценные проекты, например, лендинг с отображением погоды в реальном времени

Формирование знаний и навыков на высоком уровне достигается за счет того, что программа носит практико-ориентированный характер. Все знания, полученные во время онлайн-лекций, отрабатываются вовремя онлайн практических занятий. Система заданий во время практики организована таким образом, что сначала обучающимся предлагаются задания на формирование и отработку конкретных умений через решение алгоритмических задач. После чего учащимся предлагается выполнение учебного проекта, например, небольшой игры или полноценного лендинга.

## **Актуальность**

Актуальность программы обусловлена переходом многих компаний в Digital. Frontend-разработчик занимается разработкой клиентской части веб-приложений, что напрямую влияет на то, какое мнение о компании складывает клиент и таким образом влияет на доходы компании. Язык программирования JavaScript ежедневно адаптируется под реалии современного мира IT, сейчас при помощи языка программирования JavaScript можно создавать не только клиентскую часть проекта, но также и серверную, что дает возможность студенту при желании сменить направление своей работы.

Язык программирования JavaScript, в настоящее время, входит в десятку наиболее востребованным во всех сферах IT по данным рейтинга TIOBE.

Обучение на данной программе позволит сформировать компетенций в области решения практических задач разработки приложений с использованием языка JavaScript, применения шаблонов проектирования, работы с базами данных, применения объектно-ориентированного. Программа позволит с помощью формирования практических навыков в разработке приложений повысить интерес школьников к программированию, что повлияет на количество поступающих абитуриентов в образовательные организации среднего профессионального и высшего образования по IT-направлениям.

## **Цель программы**

Формирование знаний, умений и навыков достаточных для разработки приложений с использованием языка программирования JavaScript.

## **Задачи программы**

Программа направлена на реализацию следующих задач обучения, развития и воспитания.

### *Задачи обучения*

- Формировать и развивать компетентность в области разработки веб-приложений.
- Формировать представление о процессах веб-разработки.
- Формировать представление об архитектуре компьютеров и сети.
- Формировать представление о правила безопасной работы с информацией.
- Формировать ИКТ-компетенции.
- Формировать представление об уровне технического прогресса.

### *Задачи развития*

- Формировать умение планировать и регулировать свою деятельность.
- Способствовать формированию самонаблюдения и самооценки в процессе познавательной, творческой деятельности.
- Формировать и развивать компетентность в области работы с информацией.
- Способствовать формированию навыков сотрудничества со взрослыми и сверстниками в разных социальных ситуациях, умения избегать конфликтов. Способствовать овладению навыками адаптации в динамично развивающемся мире.

### *Задачи воспитания*

- Способствовать формированию нравственных норм и ценностей в поведении и сознании.
- Способствовать принятию и освоению социальной роли обучающегося, развитию мотивов учебной деятельности и формированию личностного смысла образования.
- Способствовать развитию самостоятельности и личной ответственности за свои поступки, в том числе в информационной деятельности, на основе представлений о нравственных нормах, социальной справедливости и свободе. Развивать рефлексивную деятельность учащихся.
- Способствовать развитию эстетических чувств, доброжелательности и эмоционально-нравственной отзывчивости, понимания чувствам других людей и сопереживания им.
- Способствовать формированию установки на безопасный, здоровый образ жизни, наличие мотивации к творческому труду, работе на результат, бережному отношению к материальным и духовным ценностям.

## **Планируемые результаты обучения**

Сформировать у обучающихся школ базовые знания и умения в области решения задач разработки приложений.

По окончании обучения по программе обучающийся будет знать:

- Понятие алгоритма и алгоритмического языка программирования.
- Назначение JavaScript и его сфера применения.
- Основы синтаксиса языка программирования JavaScript.
- Основы программирования на JavaScript:  
Базовые операторы, переменные и их типы, константы, массивы, управляющие конструкции, условные операторы, циклы, функции, принципы создание функций, аргументы функции, синхронные и асинхронные функции, область видимости переменных, стандартные библиотеки языка программирования JavaScript.
- Основы верстки HTML и CSS.
- Систему контроля версий.
- Возможности используемой системы контроля версий и вспомогательных инструментальных программных средств для обработки исходного текста программного кода.

- Основные принципы отладки и тестирования программных продуктов. Типы тестирования.
- Методы и приемы отладки программного кода. Типы и форматы сообщений об ошибках, предупреждениях.
- Принципы работы современных компиляторов, отладчиков и оптимизаторов программного кода.

По окончании обучения по программе обучающийся будет уметь:

- Устанавливать и настраивать среду разработки JavaScript.
- Применять язык программирования JavaScript для написания программного кода для решения учебных и практических задач.
- Создавать структуру кода, размещающего элементы приложения.
- Создавать макеты сайтов
- Использовать возможности языка программирования JavaScript для разработки клиентской части веб-приложений.
- Использовать HTML и CSS для верстки веб-приложений.
- Работать с системой контроля версий.
- Использовать инструментальные средства для отладки и тестирования.
- Выполнять отладку и тестирование программы на уровне модуля.
- Выявлять ошибки в программном коде.
- Интерпретировать сообщения об ошибках, предупреждения.

### **Категория обучающихся (возраст) по программе**

Учащиеся, осваивающие образовательные программы основного общего и среднего общего образования (с 8 по 11 классы)

### **Форма обучения:**

Очная форма обучения без применения дистанционных образовательных технологий.

### **Срок реализации программы:**

2 года

### **Трудоемкость программы:**

150 академических часов в рамках двухлетнего курса обучения, с включением 4 модулей трудоемкостью не менее 36 академических часов.

### **Ссылка на официальную страницу курса:**

[http://cospaces.synergyitschool.com/javascript\\_offline](http://cospaces.synergyitschool.com/javascript_offline)

## УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН

№ п/п	Наименование модулей/тем программы	Всего, час	Виды учебных занятий			Формы контроля
			Ле кц ии	П ра кт ич ес ки е за ня ти я	Са мо ст оя те ль на я ра бо та	
1	Тема 1.1. HTML и CSS	26	2	12	12	
2	Тема 1.2 Figma	10	2	2	6	
3	Тема 2.1 Практическое применение верстки	15	4	5	6	
4	Тема 2.2 Основы программирования на JavaScript	21	4	5	12	
5	Тема 3.1 Расширенные возможности JavaScript	8	1	3	4	
6	Тема 3.2 Фреймворк React	28	2	12	14	
7	Тема 4.1 Фреймворк Vue.JS	12	2	4	6	
8	Тема 4.2 Фреймворк Node.JS	18	2	6	10	
9	Тема 4.3 Тестирование приложений	6	2	2	2	
13	Промежуточная аттестация	4				Тестирование
14	Итоговая аттестация	2				Защита проекта
15	Итого	150	21	51	72	



## РАБОЧАЯ ПРОГРАММА

### Модуль 1. Введение в разработку. Верстка сайтов.

Модуль знакомит с версткой страниц для проектов при помощи HTML и CSS, основами дизайна, принципами работы с элементами на странице. Обучающиеся делают верстку адаптивной, по итогу создадут три полноценных макета.

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы
Тема 1.1. HTML и CSS	Верстка сайтов с помощью HTML и CSS, базовая структуры страницы, на изучение особенностей отображения элементов веб-приложений в различных браузерах, научиться с использовать язык разметки страниц веб-страниц	HTML и CSS, работа с медиафайлами, принципы адаптивности и кроссбраузерности, верстка HTML-писем	Повторение материалов по теме. Самостоятельная работа студента по данному модулю заключается в создании трех полноценных, адаптивных макетов для сайтов, а также закрепят навыки работы в редакторе кода
Тема 1.2 Figma	Логика верстки и создания сайтов с помощью графического редактора Figma	Работа со специализированным программным обеспечением для верстки страниц веб-приложений, таких как графического редактора Figma, работа с элементами внутри Figma и на приобретение навыка дизайн-анализа макета и насмотренности у студента	Повторение материалов по теме. Создание полноценного макета при помощи графического редактора Figma
Промежуточная аттестация			

### Модуль 2. Введение в разработку на языке JavaScript.

Модуль знакомит с синтаксисом и основами программирования на языке JavaScript: принципами объектно-ориентированного программирования, асинхронными запросами.

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы
Тема 2.1 Практическое применение верстки	Повторение материалов по основам верстки и более углубленно изучить программный код HTML и CSS	Создание полноценного адаптивного проекта, который поможет закрепить навыки анализа дизайн-макета и подключения к ИР	Повторение материалов по теме. Создание лендинга, с использованием программного кода и выученных технологий, что поможет закрепить



		стилей оформления веб-страниц	навыки верстки HTML и CSS
Тема 2.2 Основы программирования на JavaScript	Познакомятся с JavaScript, узнают для чего он нужен, область применения, а также узнают какие бывают редакторы кода, также студентам предстоит познакомиться с понятием стрелочные функции	Синтаксис интерпретируемого языка Javascript. Изучение операторов в JS, циклы, функции, объекты и массивы, в качестве своей проектной работы создадут игру "камень, ножницы, бумага", научатся решать задачи при помощи JavaScript, работать с регулярным выражениями и асинхронными запросами, promis, map и set	Повторение материалов по теме. Создание лендинга при помощи программного кода JavaScript
Промежуточная аттестация			

### Модуль 3. Продвинутый JavaScript. Работа с фреймворком React.

Модуль знакомит с расширенными возможностями JavaScript: работа с ES6, генераторы, регулярными выражениями, пакетными менеджерами и инструментами сборки, а также познакомится с инструментом - фреймворк React и создадут при помощи него новый проект

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы
Тема 3.1 Расширенные возможности JavaScript	Знакомство с принципами работы с ES6, генераторами, асинхронными запросами, пакетными менеджерами.	Использование ES6, асинхронных запросов, регулярных выражений, пакетных менеджеров и инструментов сборки для решения учебных и практических задач.	Повторение материалов по теме. Решение практических и учебных задач по теме.
Тема 3.2 Фреймворк React	В данных лекциях студенты познакомятся с пакетными менеджерами и инструментами сборки в фреймворке React	В практических заданиях студенты научатся работать с хуками, манипулировать данными и проводить Unit-тестирование	Повторение материалов по теме. Разработка веб-приложений (кинотеатр) с использованием фреймворка React
Промежуточная аттестация			

### Модуль 4. Архитектура приложений. Тестирование

Модуль знакомит с шаблонами проектирования программного обеспечения, конфигурацией, стандартами кодирования и интерфейсами. Во время обучения, обучающиеся научатся изменять базовую конфигурацию без создания опасных ситуаций, разрабатывать полноценное приложение и выполнять его тестирование.

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы

Тема 4.1 Фреймворк Vue.JS	Знакомство с целями применения фреймворка Vue.JS в проектах, познакомиться с особенностью программного кода при применении фреймворка в проектах	Работа с фреймворком Vue.JS, размещение программного кода в клиентской части веб-приложения, компонентами и хуками фреймворка и использовать различные шаблоны	Повторение материалов по теме. Создание лендинга с получением погоды в реальном времени, используя программный код и компоненты программно-технических архитектур веб-приложения
Тема 4.2 Фреймворк Node.JS	Знакомство с назначением фреймворка Node.JS и его применением в проектах	Работа с пакетными менеджерами: Npm, Yarn. Использование подаemon. Использование экспресс-роутеров для решения практических задач	Повторение материалов по теме. Создание многопользовательского масштабируемого чата.
Тема 4.3 Тестирование приложений	Тестирование приложения: моки и стабы. Непрерывная доставка Phing. JMeter нагрузочные испытания. Автотесты	Тестирование ошибок. Подготовка данных для тестов. Тестирование кода, взаимодействующего с файлами. Работа с тестами.	Повторение материалов по теме. Тестирование приложения.
Промежуточная аттестация			
Итоговая аттестация			

## ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

### Кадровое обеспечение

Фамилия	Имя	Отчество	Уровень образования	Ученая степень	Ученое звание	Наименование основного места работы	Должность	Портфолио
Сосковец	Лариса	Александровна	высшее педагогическое	нет	нет	Университет Синергия	Старший преподаватель кафедры ГиЕД	<a href="https://docs.google.com/document/d/1az089wbADo1DxX156iG8waqmYU1Wn40z/edit?usp=sharing&amp;ouid=103347011653736542142&amp;rtpof=true&amp;sd=true">https://docs.google.com/document/d/1az089wbADo1DxX156iG8waqmYU1Wn40z/edit?usp=sharing&amp;ouid=103347011653736542142&amp;rtpof=true&amp;sd=true</a>
Гедыгушева	Татьяна	Хаучьяновна	высшее	Кандидат педагогических наук		Университет Синергия	Заведующая кафедрой математических и естественно-научных дисциплин	<a href="https://disk.yandex.ru/d/RKE0HXb1mcg6gg">https://disk.yandex.ru/d/RKE0HXb1mcg6gg</a>

**Адреса и координаты**

№ п/п	Название адреса	Адрес	Код адреса	Долгота	Широта
1	Омский филиал Университета «Синергия»	г.Омск, проспект Карла Маркса, 18, корп. 10	52000000000	54.966280	73.382543
2	Карачаево- Черкесский филиал Университета «Синергия»	г.Черкесск просп. Ленина, 83	91000000000	44.217198	42.047749
3	Колледж экономики и права	г. Щелково, ул. Пионерская, д.19	46000000000	55.922706	37.969092



## **Методическое обеспечение. Методы, формы и технологии, применяемые при реализации программы**

Обучение на курсе организовано в групповой и индивидуально-групповой форме.  
Основная форма обучения: онлайн-лекции и практики, в том числе с демонстрацией экрана.

Курс разработан с использованием следующих образовательных технологий:

- технология развивающего обучения,
- технология проблемного обучения,
- проектная технология,
- кейс-технология,
- обучение в сотрудничестве,
- здоровьесберегающие технологии.

При разработке материалов курса учитываются различные типы перцептивная модальности, поэтому используются словесные, наглядные, практические методы.

Ко всем необходимым материалам доступ обучающихся организуется через платформу обучения: презентациям, заданиям, записям прошедших вебинаров.

Все знания, полученные во время онлайн-лекций, отрабатывается вовремя онлайн практических занятий. Система заданий во время практики организована таким образом, что сначала обучающимся предлагаются задания на формирование и отработку конкретных умений через решение алгоритмических задач. После чего учащимся предлагается выполнение учебного проекта, например, создание сайта-визитки, новостной страницы, калькулятора, игры, создания формы обратной связи, формы регистрации, справочника, создания схемы базы данных собственного проекта и реализации работы веб-приложения с базой данных.

## **Материально-техническое обеспечение**

- Персональный компьютер, который соответствует минимальным требованиям:

Для ПК на Windows: процессор i-серии (i3, i5, i7) или аналогичный от AMD (от 4 ядер), от 4G B RAM.

Для ПК на Mac OS: конфигурация Intel i7-2720QM/RAM 4 GB/HDD 750 GB., ОС Windows 10 (версии Home достаточно) либо Mac OS X;

- Высокоскоростной стабильный доступ в интернет.

## УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

В процессе обучения по образовательной программе система заданий организована таким образом, что сначала обучающимся предлагаются задания на формирование и отработку конкретных умений через решение алгоритмических задач. После чего учащимся предлагается выполнение учебного проекта. Ниже представлены примеры некоторых из них.

### Пример задания по Теме 1.1. HTML и CSS

Задача: ознакомиться с инструментами разработчика: где смотреть код, где отображаются ошибки, как посмотреть режим предусмотренный для разных видов устройств и т. д.

Процесс выполнения:

- 1) Откройте браузер
- 2) Выберите любой сайт на Ваш выбор
- 3) Если Вы используете Firefox или Google Chrome, то можете нажать F12 верху клавиатуры. Если у Вас другой браузер, то можете выбрать меню настроек браузера => дополнительные инструменты => инструменты разработчика
- 4) Найдите инструменты разработки в Вашем браузере и попробуйте открывать их на различных сайтах (лучше используйте Chrome или Firefox)
- 5) Зайдите во вкладку Elements и попробуйте перейти к коду какого-нибудь элемента с помощью инструмента “выбор элемента с помощью курсора”
- 6) Посмотрите, как сайт будет выглядеть на различных устройства с помощью инструмента “переключение в режим выбора устройств”

Задача: научиться подключать стили на html страницу.

Процесс выполнения:

- 1) Имеем подготовленную стандартную html страницу с минимальным кодом, которую необходимо скопировать в файл index.html в папке NewProject (создать):

```
<!DOCTYPE HTML>
```



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Документ без названия</title>

</head>
<body>
<h1>Я изучаю свойства CSS</h1>

</body>
</html>
```

2) Используем внешние стили (из внешнего файла с расширением .css). Для этого в строку, следующую за <title>Документ без названия</title> вставляем ссылку <link href="style.css" type="text/css" rel="stylesheet">. Данная ссылка будет указывать браузеру, что необходимо подключиться к внешнему файлу с расширением .css и использовать стили (шрифты, цвета шрифтов, фона, толщину шрифтов, взаимное размещение блоков относительно друг друга и пр.)

3) Создаём новый файл с именем и расширением style.css 4) В этом файле создаём код: h1 {  
color: red;  
}

4) Сохраняем файл (в командной строке “файл сохранить как”) на рабочий стол и запускаем сохранённый файл index.html.

Результат, который должен получить студент:

Открывшееся окно браузера с надписью в левом верхнем углу крупным шрифтом “Я изучаю свойства CSS” написанные шрифтом красного цвета.

### **Пример заданий по Теме 1.2 Figma**

Задача: Установить программу на компьютер, зарегистрироваться, освоить основные функции работы с Figma.

Процесс выполнения:

- 1) Зайти на сайт [figma.com](https://figma.com)
- 2) Создать новую учетную запись, либо авторизоваться через свой гугл-аккаунт
- 3) Создать “new design file”
- 4) Дать ему название “My first project by Figma”
- 5) Освоить базовые инструменты работы в Figma, такие как создание фрейма, прямоугольника, линии, текста.

Результат, который должен получить студент: Открытая веб версия Figma с созданным проектом “new design file” внутри которого созданы фрейм, прямоугольник, линия и набран любой текст.

Задача: Создать новый проект в Figma, импортировать из интернета фотографию котика, поместить его в центре веб-приложения, придать фотографии размеры 470x250. С помощью стандартных функций создать рамку для фотографии котика, раскрасить её в золотистый цвет F5E125. Подписать сверху “Кот”. Экспортировать созданный файл.

Процесс выполнения:

В открытом приложении скопировать изображение котика из интернета. С помощью стандартных настроек придать ему заданные параметры.

С помощью rectangle отрисовать рамку и присвоить ей указанный цвет.

После этого нажать на export и сохранить полученный рисунок.

Результат, который должен получить студент: Скачанный файл с изображением котика в золотистой рамке.

### **Пример заданий по Теме 2.1 Практическое применение верстки**

1. Напишите, с помощью какого кода подключаются медиа-запросы? Что вообще такое – медиа-запрос?
2. Что означают breakpoint?
3. Напишите код, отражающий медиа-запрос под максимальную ширину экрана 500px.
4. Откройте в браузере созданный ранее сайт и поэкспериментируйте в преобразовании элементов экрана через инструмент разработчика, используя материалы лекции спикера.

### **Пример задания по Теме 2.2 Основы программирования на JavaScript**

Задача: Практическое применение знаний, полученных из теоретической части курса.

Подготовка и начало выполнения создания макета на figma.

Процесс выполнения:

- 1) Зайти на сайт [figma.com](https://figma.com)
- 2) Создать новую учетную запись, либо авторизоваться через свой гугл-аккаунт
- 3) Внимательно повторять за спикером всю последовательность действий по формированию проекта.

Результат, который должен получить студент: Начать создавать проект под руководством спикера.

Задача: Понять как создается сайт на нескольких страниц. Способы подключения стилей к каждому из html файлу. Изучение специфических тегов.

Процесс выполнения:

- 1) Открыть VSC.
- 2) Создать файлы index.html и style.css main.html и main.css.
- 3) Произвести привязку стилей с гипертекстовой разметкой с помощью `<link rel="stylesheet" href="style.css, либо main.css">` в зависимости от того, какой файл гипертекста вы привязываете.
- 4) Внимательно повторять за спикером всю последовательность действий по формированию проекта.

### **Пример задания по Теме 3.1 Расширенные возможности JavaScript**

Задачи: Узнать в чём отличие нового JS, особенности задания переменных.

1. Напишите отличие переменных var и let.
2. Переменная const. Чем характерна? Что выведет терминал при выполнении следующего кода:

```
const c = 7;
```

```
c = -7;
```

```
console.log(c);
```

3. Возможно ли всё таки изменить значение константы?
4. Напишите код, где задаётся функция sum в которой прописывается суммирование двух значений a и b. Вывести в терминал.
5. Стрелочные функции. Напишите пример умножения числа a на число b через стрелочную функцию в одну строку! Умножьте 11 на 8.

## Пример проекта по Теме 3.2 Фреймворк React

Задача: Понять что такое хуки и как ими пользоваться.

Процесс выполнения:

- 1) Создать в Visual Studio Code IDE в терминале проект “hooks” по стандартному шаблону: `npx create-react-app hooks`
- 2) Нажать `enter` и некоторое время подождать, пока приложение создастся.
- 3) Работаем с файлом `App.js`, который является нашим функциональным компонентом.
- 4) Задача- на основе лекции спикера написать упрощенную версию счетчика, который будет только увеличивать значения на единицу.
- 5) Изначально импортировать метод из React (`import React, {useState} from 'react'`)
- 6) Вызываем его в функции (соблюдая все правила, изученные ранее):

```
import React, {useState} from 'react'
```

```
function Count() {
```

```
  const [count, setCount] = useState(1)
```

```
  return (
```

```
    <div>
```

```
      Count
```

```
    </div>
```

```
  )
```

```
}
```

7) Хук `useState()` принимает один параметр в виде начального состояния, которое может быть представлено как значение (строка, число, массив, объект и т.д.) или функция. После вызова функции, `useState` возвращает массив из двух элементов. Первым элементом является начальное состояние, которое мы передаём, вторым — функция, изменяющая это состояние при ее вызове и передаче нового значения:

```
import React, {useState} from 'react'
```

```
function Count() {
```

```
  const [count, setCount] = useState(1)
```

```
  const increment = () => {
```

```
    setCount(count + 1)
```

```
  }
```

```
  return (
```

```
    <div>
```

```
      <button onClick={increment}>+</button>
```

```
      <span>
```

```
        {count}
```

```
      </span>
```

```
</div>
```

```
)
```

```
}
```

Результат, который должен получить студент: Реальная задача, которую задают на собеседованиях- получаем простейший счетчик. При нажатии на кнопку, значение счетчика увеличивается на единицу.

## ФОРМЫ АТТЕСТАЦИИ

Аттестация проводится в двух формах: промежуточная и итоговая.

Промежуточная аттестация. Данный вид аттестации проводится по итогам освоения каждого модуля.

Форма контроля - тестирование.

Показатели и критерии оценивания:

Оценка теста производится автоматически. Тест считается пройденным, если даны правильные ответы не менее, чем на 50% тестовых заданий.

### Шкала оценивания

Нижнее значение	0%
Верхнее значение	100%
Минимальный проходной балл для успешной сдачи	50%

Итоговая аттестация. Данный вид аттестации проводится по итогам освоения всей Образовательной программы, включая освоение ее последнего модуля.

Форма контроля – защита проектов.

Показатели и критерии оценивания:

Проекты проверяются в соответствии с заданиями, описанными в материалах.

За каждое выполненное условие ставится от 0 до 2 баллов согласно критериям оценивания.

Критерии оценивания:

0 — условие не выполнено

1 — условие выполнено, частично

2 — условие выполнено, полностью

### Шкала оценивания

Нижнее значение	0
Верхнее значение	10
Минимальный проходной балл для успешной сдачи	5



## ОЦЕНОЧНЫЕ МАТЕРИАЛЫ, ВКЛЮЧАЯ ПРИМЕРЫ КОНТРОЛЬНЫХ ЗАДАНИЙ

Для каждого вида аттестации предусмотрены свои оценочные материалы. Ниже представлены примеры некоторых из них.

### Промежуточная аттестация по модулю 1

1. Что такое флексбоксы?
  - а) Это блоки стандартного размера, в которых содержится контекст сайта. Пришли взамен таблицам, на которых ранее верстались все сайты без исключения
  - б) Флексбокс - это внутренняя структура HTML страницы
  - в) Флексбоксы, так называемые гибкие блоки, дают веб-разработчикам контроль над расположением элементов, их выравниванием и размерами внутри контейнера
  
2. Какой из нижеприведённых кодов создаёт флекс контейнер?
  - а) 

```
.container {  
    display: flex;  
}
```
  - б) 

```
.container {  
    flex - grow: 1;  
}
```
  - в) 

```
.container > div {  
    font-size: 5vw;  
    padding: .5em;  
    color: white;  
}
```

3. Если создать 3 флекс контейнера одинаковых размеров и начать один из них наполнять контентом, что произойдёт, когда этого контента станет слишком много?
- а) Заполняемый контейнер разделится на 2 контейнера, текст разобьётся равномерно на эти два контейнера
  - б) Текст будет постепенно перемещаться на соседние контейнеры
  - в) Заполняемый контейнер по мере наполнения будет вытеснять пространство у двух пустых соседних контейнеров
4. Как называется тип вёрстки, где используются флексбоксы?
- а) Резиновый тип
  - б) Адаптивный тип
  - в) Фиксированный тип
5. Для чего в CSS используют трансформации?
- а) Для того, чтобы веб страница смотрелась одинаково читабельно на любых мониторах
  - б) Для изменения вида, поворота и других преобразований элемента
  - в) Для преобразования элементов css кода в двоичный код
6. Для чего нужны циклы в js?
- а) Для того, чтобы сделать постоянным и устойчивым соединением (зациклить) html и css файлов
  - б) Циклы позволяют перелистывать страницы сайта более плавно, как бы по кругу (циклом)
  - в) Для того, чтобы выполнить однотипное действие заданное количество раз
7. Какие значения будут выведены при выполнении следующего цикла:
- ```
let i = 0;
```
- ```
while (++i < 5) alert( i );
```

а) 0 1 2 3

б) 1 2 3 4 (Первое значение:  $i = 1$ , так как операция  $++i$  сначала увеличит  $i$ , а потом уже произойдёт сравнение и выполнение `alert`).

Далее 2, 3, 4... Значения выводятся одно за другим. Для каждого значения сначала происходит увеличение, а потом – сравнение, так как  $++$  стоит перед переменной.

При  $i = 4$  произойдёт увеличение  $i$  до 5, а потом сравнение `while (5 < 5)` – это неверно. Поэтому на этом цикл остановится, и значение 5 выведено не будет.)

в) 1 2 3 4 5

8. Какие значения будут выведены при выполнении следующего цикла:

```
let i = 0;
```

```
while (i++ < 5) alert( i );
```

а) 0 1 2 3

б) 1 2 3 4

в) 1 2 3 4 5

### **Промежуточная аттестация по модулю 2**

1. За что отвечает “вёрстка сайта”?

а) Расположение контента на странице

б) Визуальные эффекты

в) Кроссбраузерность

2. Что такое CMS система?

а) Система управления кроссбраузерностью сайта

б) Управление всеми элементами ресурса через административную часть

в) Система взаимосвязи между пользователем и сервером

3. Браузер - это...

а) Страница, на которой отображается DOM

б) Программа, которая воспроизводит контент с вэб ресурсов

в) Система управления стилями вэб страницы

4. HTML - это:

а) Язык стилей. С помощью него создаются динамичные и яркие страницы сайта.

б) Язык гипертекстовой разметки.

в) Серверный язык, представленный в виде двоичного машинного кода.

5. Что такое гипертекст?

а) Это ссылки, которые соединяют вэб страницы между собой.

б) Это шаблон любого сайта

в) Это текстовое наполнение вэб страницы

6. Что выведет следующий код?

`<h1>Text</h5>`

а) Text в виде главного заголовка h1

б) Text в виде справочного заголовка h5

в) Ошибку

7. Какие теги выделяют текст жирным шрифтом?

а) `<i>` `<em>`

б) `<b>` `<strong>`

в) `<br\>` `<hr\>`

8. Что такое Figma?
- а) Это оффлайн приложение, используемое для работы с таблицами
  - б) Figma – онлайн-платформа для разработки графических проектов: логотипов, интерфейсов, иллюстраций
  - в) Figma – облачное хранилище кода для использования одним или несколькими программистами
9. Что такое фрейм?
- а) Фрейм – полностью оформленный документ, например, страница сайта
  - б) Фрейм - это шаблон HTML документа
  - в) Это программа, позволяющая писать различные приложения, сайты

### **Промежуточная аттестация по модулю 3**

1. Что делает Rest-оператор?
- а) Позволяет анимировать элементы при трансформации. Это экономит память и сокращает время выполнения кода
  - б) Позволяет "свернуть" часть элементов во время деструктуризации. Например, с его помощью можно разложить массив на первый элемент и все остальные
  - в) Вызывает задержку выполнения функции
2. Что означает следующая запись?
- ```
function doSomething(first, second, ...rest) {  
    // do something cool  
}
```
- а) Функция doSomething будет выполняться следующим образом: Первые два параметра начнут выполняться только при выполнении условия true, которое будет ответом на передачу третьего и последующих параметров
  - б) Функция doSomething будет выполняться следующим образом: Первые два параметра начнут выполняться только при выполнении условия false, которое будет ответом на передачу третьего и последующих параметров
  - в) Можно получить доступ к первым двум именованным параметрам, как и обычно. А вот уже к другим аргументам, переданным функции начиная с третьего, вы сможете получить доступ через массив под названием rest, куда все они автоматически будут собираться
3. Что будет на выходе, если на вход предыдущего примера подать меньше чем три параметра?
- а) Команда остановится и выдаст ошибку компиляции

- б) Команда выполнится только с двумя первыми заданным аргументами
- в) rest будет просто пустым массивом

4. Что такое Map?

- а) Это аналог object, просто в связи с расширением синтаксиса потребовалось ввести новые значения некоторым функциям, параметрам, переменным (как, например, var и let)
- б) Map – это коллекция ключ/значение, как и Object. Но основное отличие в том, что Map позволяет использовать ключи любого типа
- в) Map – это особый вид коллекции: «множество» значений (без ключей), где каждое значение может появляться только один раз

5. Расскажите об объекте Set?

- а) Объект Set – это особый вид коллекции: «множество» значений (без ключей), где каждое значение может появляться только один раз
- б) Объект Set – это особый вид коллекции: «множество» ключей, где каждый ключ может появляться только один раз
- в) По сути, Set это словарь, в котором обращение к каждому элементу возможно единожды, после чего он принимает нулевое значение

6. Что такое генераторы?

- а) Это новый вид функций, которые позволяют содержать в себе ещё набор других функций и запускать их по мере необходимости (либо не запускать)
- б) Это функции, которые, при определённых параметрах, создают новые функции, циклы
- в) Генераторы – новый вид функций в современном JavaScript. Они отличаются от обычных тем, что могут приостанавливать своё выполнение, возвращать промежуточный результат и далее возобновлять его позже, в произвольный момент времени

7. Приведите синтаксис создания генератора.

а) 

```
function generateSequence() {  
  yield 1;  
  yield 2;  
  return 3;  
}
```

б) 

```
function* generateSequence() {  
  yield 1;  
  yield 2;  
  return 3;  
}
```

```
в) def function generateSequence() {  
  yield 1;  
  yield 2;  
  return 3;  
}
```

#### **Промежуточная аттестация по модулю 4**

1. Что есть Vue.js?

- а) Vue.js - пакетный менеджер для среды NodeJS. Используют для установки Gulp и его плагинов
- б) Это фреймворк для создания серверной составляющей сайта
- в) Популярный фронтенд фреймворк

2. Опишите основные концепции Vue.js

- а) конструктор, компоненты, директивы, переходы
- б) конструктор, совместимость, библиотеки, переходы
- в) библиотеки, функции, переходы, безопасность

3. Что такое компоненты Vue.js?

- а) Всё из чего состоит Vue.js - поля, ссылки и т.д. - это компоненты
- б) Компоненты — это переиспользуемые экземпляры Vue со своим именем
- в) Компоненты - это библиотеки, используемые в фреймворке по умолчанию

4. Что такое Vue Hooks?

- а) Это приложение, которое позволяет визуализировать эффекты на вэб сайтах

б) Понятие хуков тесно связано с жизненным циклом хуков. Хуки создания, обновления, установки и уничтожения компонентов - это все циклы, которые проходят компоненты при любых действиях

в) Хуки - это компоненты фреймворка, которые отражают процессы инициализации элементов сайта

5. Что такое Nuxt.js?

а) Это библиотека в составе Node.js

б) Nuxt.js - это фреймворк для создания универсальных приложений на Vue.js с использованием Node.js

в) Это приложение, связывающее серверную и frontend части сайта

6. Назовите наиболее известные эмуляторы терминала

а) Mac OS X: iTerm 2

Windows: PuTTY

Linux: Terminal

б) Mac OS X: Terminal

Windows: PuTTY

Linux: Term 2

в) Mac OS X: iTerm 2

Windows: Terminal

Linux: PuTTY



7. GIT - это:

- а) Это централизованное хранилище информации, куда имеет доступ с правами просмотра каждый желающий. На этом сервере может храниться любая информация, любой тип данных
- б) Git — распределённая система управления версиями. Она не имеет выделенного сервера
- в) Фреймворк для frontend разработчиков

### **Итоговая аттестация**

Для прохождения итоговой аттестации необходимо предоставить самостоятельно разработанное веб-приложение. На экспертную оценку необходимо направить презентацию, содержащую информацию о вашем проекте и его компонентах. Проект должен соответствовать следующим требованиям:

- 1) Проект должен быть рабочим, без зависаний, лагов, соответствовать его описанию
  - 2) Должно присутствовать его текстовое описание (приложен файл tutorial.txt)
  - 3) Проект должен быть размещён на Github
  - 4) Код должен быть написан в строгом соответствии с принятыми стандартами соответствующего языка программирования, то есть, соблюден синтаксис, отступы, переменные должны иметь названия, логически отражающие контекст программы.
  - 5) Код должен быть рационален - то есть написан максимально оптимально с точки зрения объема, выполняться насколько это возможно быстро, не содержать бесконечных циклов.
- Внутри кода обязательно должны присутствовать комментарии, объясняющие ключевые моменты
  - Если проект содержит несколько файлов, то их названия должны иметь соответствующие расширения и их имена должны логически соответствовать их содержанию (например, если в файле содержится код на языке javascript и он отвечает за выполнение математических вычислений, то название должно быть похожим на math.js и так далее).

## **ПЕРЕЧЕНЬ ИСТОЧНИКОВ ИНФОРМАЦИОННОГО СОПРОВОЖДЕНИЯ**

- Элизабет Робсон, Эрик Фримен Изучаем HTML, XHTML и CSS. 2-е изд. — СПб.: Питер, 2014. — 720 с.: ил. — (Серия «Head First O'Reilly»)
- Роберт Мартин Чистый код. Создание, анализ и рефакторинг Роберт Мартин СПб.: Питер, 2018.
- Никсон Робин. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 – 5-е изд. – СПб.: Питер, 2019. - 816 с.

## КОМПЕТЕНЦИИ, ФОРМИРУЕМЫЕ В ПРОЦЕССЕ ОБУЧЕНИЯ

| Наименование компетенции                                                       | Тип компетенции      | Знания, соответствующие компетенции                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Умения, соответствующие компетенции                                                                                                                                                                                                                                                                                                                    | Владение инструментами, соответствующие компетенции |
|--------------------------------------------------------------------------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| Способен создавать программы (кодировать) на языке программирования JavaScript | общепрофессиональные | <p>Понятие алгоритма и алгоритмического языка программирования.</p> <p>Назначение JavaScript и его сфера применения.</p> <p>Основы синтаксиса языка программирования JavaScript.</p> <p>Особенности программирования на языке JavaScript.</p> <p>Базовые операторы.</p> <p>Переменные и их типы.</p> <p>Константы.</p> <p>Массивы.</p> <p>Управляющие конструкции.</p> <p>Условные операторы.</p> <p>Операторы инкремента, декремента.</p> <p>Циклы.</p> <p>Функции.</p> <p>Создание функции.</p> <p>Аргументы функции и аргументы функции по умолчанию.</p> <p>Область видимости переменных.</p> <p>Статические переменные.</p> <p>Локальные и глобальные переменные.</p> <p>Стандартные библиотеки языка программирования JavaScript</p> <p>Синхронные и асинхронные функции.</p> <p>Основы работы с фреймворками React, Node.js, Vue.js, NuxtJS</p> | <p>Установка и настройка среды разработки JavaScript.</p> <p>Применять язык программирования JavaScript для написания программного кода для решения учебных и практических задач.</p> <p>Применение фреймворков React, Node.js, Vue.js, NuxtJS</p> <p>Разрабатывать программный код приложений с использованием языка программирования JavaScript.</p> | - Средствами разработки: Visual Studio Code         |
| Умеет верстать страниц веб-приложений                                          | общепрофессиональные | <p>Знает особенности отображения элементов IP в различных браузерах, особенности отображения IP в размерах рабочего пространства устройств, методы повышения</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <p>Умеет применять нормативные документы, определяющие требования к оформлению страниц IP, определять возможности отображения web-страниц в размерах рабочего</p>                                                                                                                                                                                      | Средствами разработки: VisualStudioCode             |

|                                                            |                      |                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                              |                                                                                                                                                                        |
|------------------------------------------------------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                            |                      | <p>читаемости программного кода, синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования, отраслевая нормативная техническая документация</p>                                                                                                         | <p>пространства устройств для разных видов дизайн-макетов, применять специализированное программное обеспечение для верстки страниц ИР, использовать язык разметки страниц ИР</p>                                                                                                            |                                                                                                                                                                        |
| Способен осуществлять проверку и отладку программного кода | общепрофессиональные | <p>Основные принципы отладки и тестирования программных продуктов.<br/>         Типы тестирования.<br/>         Методы и приемы отладки программного кода.<br/>         Типы и форматы сообщений об ошибках, предупреждения.<br/>         Принципы работы современных компиляторов, отладчиков и оптимизаторов программного кода</p> | <p>Выполнять отладку и тестирование программы на уровне модуля.<br/>         Выявлять ошибки в программном коде.<br/>         Интерпретировать сообщения об ошибках, предупреждения.<br/>         Использовать инструментальных средств для отладки и тестирования программного продукт.</p> | <p>- Средствами разработки: Visual Studio Code, PyCharm<br/>         - Командной строкой (cmd)<br/>         - Инструментом нагрузочного тестирования Apache JMeter</p> |
| Способен работать с системой контроля версий Git           | общепрофессиональные | <p>Система контроля версий. Интерфейс Git.<br/>         Возможности используемой системы контроля версий и вспомогательных инструментальных программных средств для обработки исходного текста программного кода.</p>                                                                                                                | <p>Применение системы контроля версий для обработки исходного текста программного кода. Регистрация изменений исходного текста программного кода в системе контроля версий</p>                                                                                                               | <p>- GitHub</p>                                                                                                                                                        |

