

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО «СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



УТВЕРЖДАЮ

Директор НОЦ «Институт  
непрерывного образования»

Е.В. Мошкина

2024 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА  
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

**«Разработка веб-приложений»**

Красноярск 2024

# 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

## 1.1. Аннотация программы

Дополнительная профессиональная программа (программа профессиональной переподготовки) для не ИТ-профиля «Разработка веб-приложений» (далее — Программа) разработана в соответствии с нормами Федерального закона РФ от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»; с учетом требований приказа Минобрнауки России от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам», с изменениями, внесенными приказом Минобрнауки России от 15 ноября 2013 г. № 1244 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499»; приказа Министерства образования и науки РФ от 23 августа 2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»; паспорта федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации»; постановления Правительства Российской Федерации от 13 мая 2021 г. № 729 «О мерах по реализации программы стратегического лидерства «Приоритет-2030» (в редакции постановления Правительства Российской Федерации от 14 марта 2022 г. № 357 «О внесении изменений в постановление Правительства Российской Федерации от 13 мая 2021 г. № 729»); приказа Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 28 февраля 2022 г. № 143 «Об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации» и признании утратившими силу некоторых приказов Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации»; федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата), утвержденного приказом Минобрнауки России от 19 сентября 2017 г. № 926, (далее — ФГОС ВО), а также профессионального стандарта 06.035 «Разработчик Web и мультимедийных приложений», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 18 января 2017 года N 44н.

Профессиональная переподготовка заинтересованных лиц (далее — Слушатели), осуществляемая в соответствии с Программой, имеющей отраслевую направленность «Информационно-коммуникационные технологии», проводится в ФГАОУ ВО «Сибирский федеральный университет»

(далее — Университет) в соответствии с учебным планом в очно-заочной форме обучения.

Разделы, включенные в учебный план Программы, используются для последующей разработки календарного учебного графика, учебно-тематического плана, рабочих программ модулей (дисциплин), оценочных и методических материалов. Перечисленные документы разрабатываются Университетом самостоятельно, с учетом актуальных положений законодательства об образовании, законодательства в области информационных технологий и смежных областей знаний ФГОС ВО и профессионального стандарта 06.035 «Разработчик Web и мультимедийных приложений».

Невозможно представить ни одну предметную область, где не было бы необходимости создания собственного веб-приложения. В ходе обучения студенты овладеют инструментами по созданию таких приложений. Это может быть лендинг для работы с целевой аудиторией. Например, создание визитки для практикующего юриста или психолога. Веб-приложения используются в сфере коммерции. С их помощью можно моментально обрабатывать запросы клиента, рассказывать об акциях или развивать систему лояльности. После прохождения обучения слушатель сможет самостоятельно разрабатывать веб-приложения, используя возможности классических языков HTML и CSS, а также популярный фреймворк для создания пользовательских интерфейсов Vue.js.

## **1.2. Цель программы**

Целью программы ДПП является формирование у слушателей, обучающихся по специальностям и направлениям подготовки, не отнесенным к ИТ-сфере, согласно приложению к Методике расчета показателя «Количество принятых на обучение по программам высшего образования в сфере информационных технологий за счет бюджетных ассигнований федерального бюджета (нарастающим итогом, начиная с 2021 года)», утвержденной приказом Минцифры России от 28 февраля 2022 г. № 143, цифровых компетенций в области информационных технологий, а именно применение языков программирования для решения профессиональных задач, разработка различных веб-архитектур, применение систем управления проектами и задачами, а также приобретение по итогам прохождения программы ДПП новой квалификации «Специалист по созданию и сопровождению информационных ресурсов».

Целевая группа: слушатели, относящиеся к категории обучающихся по специальностям и направлениям подготовки, не отнесенным к ИТ-сфере.

## **1.3. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и(или) уровней квалификации**

**1.3.1. Область профессиональной деятельности** слушателя, прошедшего обучение по программе профессиональной переподготовки, в которой может осуществлять профессиональную деятельность: проектирование, разработка и интеграция информационных ресурсов в локальной сети и информационно-телекоммуникационной сети «Интернет».

Выпускники могут осуществлять профессиональную деятельность в других областях и(или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника.

**1.3.2. Объекты профессиональной деятельности:** информационные ресурсы, веб-приложения.

Виды профессиональной деятельности: создание, модификация и сопровождение web-сайтов, корпоративных порталов организаций, мультимедиа и интерактивных приложений, информационных ресурсов.

**1.3.3. Уровень квалификации.** В соответствии с приказом Минтруда России от 18.01.2017 N 44н «Об утверждении профессионального стандарта «Разработчик Web и мультимедийных приложений» (зарегистрировано в Минюсте России 31.01.2017 N 45481), дополнительная профессиональная программа профессиональной переподготовки «Разработка веб-приложений» обеспечивает достижение шестого уровня квалификации.

**1.3.4. Компетенции (трудовые функции) в соответствии с профессиональным стандартом (формирование новых или совершенствование имеющихся)**

Программа разработана в соответствии с актуальными квалификационными требованиями, профессиональными стандартами специалистов. Виды профессиональной деятельности, трудовые функции, указанные в профессиональном стандарте 06.035 «Разработчик Web и мультимедийных приложений», представлены в таблицах 1–2.

**Характеристика новой квалификации, связанной с видом профессиональной деятельности и трудовыми функциями в соответствии с профессиональным стандартом 06.035 «Разработчик Web и мультимедийных приложений»**

Трудовые действия	Трудовая функция	Обобщенная трудовая функция	Вид профессиональной деятельности
Анализ и проверка исходного программного кода	А/01.3 Проверка и отладка программного кода	А Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов	Проектирование, разработка и интеграция информационных ресурсов в локальной сети и информационно-телекоммуникационной сети «Интернет»
Составление формализованных описаний решений поставленных задач в соответствии с требованиями принятыми в организации нормативных документов	С/01.6 Анализ и формализация требований к ИР	С Управление работами по созданию (модификации) и сопровождению информационных ресурсов	
Разработка, изменение архитектуры ИР, согласование с системным аналитиком и архитектором	С/03.6 Проектирование ИР		

**Характеристика новой и развиваемой цифровой компетенции в ИТ-сфере, связанной с уровнем формирования и развития в результате освоения программы «Разработка веб-приложений»**

Наименование сферы	Наименование профессиональной компетенции	МИНИМАЛЬНЫЙ ИСХОДНЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ  Способность не проявляется/ проявляется в степени, недостаточной для отнесения к 1 уровню сформированности компетенции	БАЗОВЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ  Способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами	ПРОДВИНУТЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ  Способность проявляется, но обучающийся эпизодически прибегает к экспертной консультации/ самостоятельно подбирает и пользуется готовыми продуктами	ЭКСПЕРТНЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ  Способность проявляется системно, на экспертном уровне / обучающийся модифицирует способность под определенные задачи / создает новый продукт, обучает других
Средства программной разработки	Применяет языки программирования для решения профессиональных задач		Применяет языки программирования для решения профессиональных задач под контролем более опытных специалистов		
Интернет-технологии	Разрабатывает различные веб-ориентированные решения		Участствует под контролем опытных специалистов в разработке веб-решений, с применением фреймворков и платформ		

#### **1.4. Планируемые результаты обучения**

Слушатели в результате освоения программы профессиональной переподготовки «Разработка веб-приложений» смогут:

PO1. Применяет языки программирования для решения профессиональных задач.

PO2. Разрабатывает различные веб-архитектуры.

PO3. Применяет методы анализ и проверка исходного программного кода.

PO4. Применять методы и средства проектирования программного обеспечения.

PO5. Применяет системы управления проектами и задачами.

#### **1.5. Категория слушателей**

Лица, получающие высшее образование по очной (очно-заочной) форме, лица, освоившие основную профессиональную образовательную программу (далее — ОПОП ВО) бакалавриата, в объеме не менее первого курса (бакалавры 2-го курса), ОПОП ВО специалитета — не менее первого и второго курсов (специалисты 3-го курса), обучающиеся по ОПОП ВО, НЕ отнесенным к ИТ-сфере.

#### **1.6. Требования к уровню подготовки поступающего на обучение**

Среднее профессиональное или высшее образование, или осваивать его в момент обучения на данной программе.

#### **1.7. Продолжительность обучения**

256 часов, из них 128 контактных, в т.ч. 16 часов стажировка.

#### **1.8. Форма обучения**

Очно-заочная (обучение по программе реализовано в формате смешанного обучения, с применением электронного обучения и дистанционных образовательных технологий).

#### **1.9. Требования к материально-техническому обеспечению, необходимому для реализации дополнительной профессиональной программы профессиональной переподготовки (требования к аудитории, компьютерному классу, программному обеспечению)**

Обучение производится на платформе электронного обучения СФУ «e-Курсы» (<https://e.sfu-kras.ru/>). Используются сервисы вебинаров и видеоконференций.

При проведении лекций, практических занятий, самостоятельной работы слушателей и стажировки используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой, высокоскоростное подключение к Интернет (не менее 5 Мбит/с).

Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Microsoft Visual Code, текстовый редактор Atom (лицензия MIT), Microsoft PowerPoint.

### **1.10. Особенности (принципы) построения дополнительной профессиональной программы профессиональной переподготовки**

Особенности построения программы переподготовки «Разработка веб-приложений»:

- в основу проектирования программы положен компетентностный подход;
- выполнение учебных заданий, требующих практического применения знаний и умений, полученных в ходе изучения логически связанных дисциплин;
- выполнение итоговых аттестационных работ по реальному заданию;
- использование информационных и коммуникационных технологий, в том числе современных систем технологической поддержки процесса обучения, обеспечивающих комфортные условия для обучающихся, преподавателей;
- применение электронных образовательных ресурсов (дистанционное, электронное, комбинированное обучение и пр.).

В поддержку дополнительной профессиональной программы профессиональной переподготовки в срок до 01.05.2023 г. будет разработан электронный курс: <https://e.sfu-kras.ru/course/view.php?id=35608>.

### **1.11. Особенности организации стажировки**

Стажировка слушателей дополнительной профессиональной программы переподготовки «Разработка веб-приложений» является обязательной составной частью образовательной программы и представляет собой вид учебной деятельности, непосредственно ориентированный на профессионально-практическую подготовку слушателей. Стажировка осуществляется в целях формирования и закрепления профессиональных умений и навыков, полученных в результате теоретической подготовки.

Сроки проведения стажировки устанавливаются графиком учебного процесса в объеме 16 часов в конце процесса обучения в соответствии с утвержденным в установленном порядке учебно-тематическим планом.

В рамках очно-заочной формы обучения на основе дистанционных технологий стажировка осуществляется в форме online стажировки (в формате разработки веб-приложения).

**1.12. Документ об образовании:** диплом о переподготовке установленного образца.



**УЧЕБНЫЙ ПЛАН**  
**дополнительной профессиональной программы профессиональной переподготовки**  
**«Разработка веб-приложений»**

Форма обучения – очно-заочная.

Срок обучения – 256 часов.

№ п/п	Наименование дисциплин	Общая трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Формы контроля
				Лекции	Лабораторные работы	Практические и семинарские занятия		
1.	Верстка веб-приложений	60	30	10		20	30	Зачет
2.	Основы языка программирования JavaScript для создания интерактивных веб-страниц	96	48	16		32	48	Зачет
3.	Разработка интерактивных веб-интерфейсов	60	30	20		20	30	Зачет
4.	Стажировка	16	12			12	4	Зачет
5.	Итоговая аттестация	24	8			8	16	Защита итоговой аттестационной работы (проекта)
	<b>Итого</b>	<b>256</b>	<b>128</b>	<b>36</b>		<b>96</b>	<b>128</b>	

**УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**  
**дополнительной профессиональной программы профессиональной переподготовки**  
**«Разработка веб-приложений»**

Категория слушателей: лица, имеющие или получающие среднее профессиональное или высшее образование.

Срок обучения: 256 часов.

Форма обучения: очно-заочная.

Режим занятий: 6 часов в неделю.

№ п/п	Наименование дисциплин	Общая трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Результаты обучения
				Лекции	Лабораторные работы	Практ. и семинарские занятия		
<b>1</b>	<b>Верстка веб-приложений</b>	<b>60</b>	<b>30</b>	<b>10</b>		<b>20</b>	<b>36</b>	<b>PO1–PO3</b>
1.1	Введение в язык гипертекстовой разметки (HTML)	6	3	1		2	3	PO1–PO3
1.2	Введение в CSS	6	3	1		2	3	PO1–PO3
1.3	CSS-селекторы	12	6	2		4	6	PO1–PO3
1.4	Блочная верстка	12	6	2		4	6	PO1–PO3
1.5	At-правила в CSS	12	6	2		4	6	PO1–PO3
1.6	Адаптивная верстка	12	6	2		4	6	PO1–PO3
<b>2</b>	<b>Основы языка программирования JavaScript для создания интерактивных веб-страниц</b>	<b>96</b>	<b>48</b>	<b>16</b>		<b>32</b>	<b>48</b>	<b>PO1–PO3</b>
2.1	Основы JavaScript	12	6	2		4	6	PO1–PO3
2.2	Primitive Value	12	6	2		4	6	PO1–PO3
2.3	Преобразование типов	12	6	2		4	6	PO1–PO3
2.4	Выражение и операторы	12	6	2		4	6	PO1–PO3
2.5	Функции	12	6	2		4	6	PO1–PO3
2.6	Объекты	12	6	2		4	6	PO1–PO3

№ п/п	Наименование дисциплин	Общая трудоем- кость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Результаты обучения
				Лекции	Лабораторные работы	Практ. и семинарские занятия		
2.7	Массивы	12	6	2		4	6	PO1–PO3
2.8	Работа с DOM (Document Object Model)	12	6	2		4	6	PO1–PO3
<b>3</b>	<b>Разработка интерактивных веб-интерфейсов</b>	<b>60</b>	<b>30</b>	<b>10</b>		<b>20</b>	<b>36</b>	<b>PO1–PO4</b>
3.1	JavaScript фреймворк Vue.js	6	3	1		2	3	PO1–PO4
3.2	Шаблоны	6	3	1		2	3	PO1–PO4
3.3	События	12	6	2		4	6	PO1–PO4
3.4	Компоненты	12	6	2		4	6	PO1–PO4
3.5	Взаимодействие с CSS	12	6	2		4	6	PO1–PO4
3.6	Анимация	12	6	2		4	6	PO1–PO4
<b>6</b>	<b>Стажировка</b>	<b>16</b>	<b>12</b>			<b>12</b>	<b>4</b>	<b>PO1–PO5</b>
<b>7</b>	<b>Итоговая аттестация</b>	<b>24</b>	<b>8</b>	<b>-</b>		<b>8</b>	<b>16</b>	<b>PO1–PO5</b>
	<b>Всего</b>	<b>256</b>	<b>128</b>	<b>32</b>		<b>96</b>	<b>128</b>	



## **II. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ**

### **2.1. Формы аттестации, оценочные материалы, методические материалы**

Программа предусматривает проведение текущей и итоговой аттестации. Текущая аттестация слушателей проводится по дисциплинам на основе выполнения заданий в электронном обучающем курсе, а также с учетом результатов промежуточного ассесмента.

Методические материалы, необходимые для выполнения текущих заданий, представлены в соответствующих элементах электронного обучающего курса и включают описание задания, методические рекомендации по его выполнению, критерии оценивания.

### **2.2. Требования и содержание итоговой аттестации**

К итоговой аттестации допускаются слушатели, успешно прошедшие процедуру итогового ассесмента. Итоговая аттестация по программе включает защиту итоговой аттестационной работы (ИАР) в форме проекта.

Основная цель итоговой аттестационной работы — выполнить работу, демонстрирующую уровень подготовленности к самостоятельной профессиональной деятельности.

ИАР может быть выполнена индивидуально или в группах. Выполнение ИАР дает возможность продемонстрировать уровень приобретенных слушателем профессиональных компетенций.

Слушатель предоставляет результат выполненной работы в формате PDF, оформленной и отвечающей требованиям к содержанию итоговой аттестационной работы. Список использованных источников литературы приводится в конце ИАР. Документ прикрепляется в организационный электронный курс программы профессиональной переподготовки «Разработка веб-приложений».

#### **Критерии оценивания итоговой аттестационной работы**

По результатам итоговых аттестационных испытаний, включенных в итоговую аттестацию, выставляются отметки по четырехбалльной системе («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»).

При осуществлении оценки уровня сформированности компетенций, умений и знаний обучающихся и выставлении отметки целесообразно использовать аддитивный принцип (принцип «сложения»):

– «отлично» выставляется слушателю, если: глубоко и прочно усвоил программный материал, использует в работе материал разнообразных литературных источников, раскрыл в полном объеме результаты проделанной работы;

– «хорошо» выставляется слушателю, если: твердо знает материал, грамотно и по существу изложен процесс выполнения работы, отчетный документ содержит достаточно необходимой информации для подтверждения выполнения проекта;

– «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, работа выполнена не в полном объеме;

– «неудовлетворительно» выставляется обучающемуся, если: студент не знает значительной части программного материала, допускает существенные ошибки, изложенные результаты не позволяют сделать вывода, что работа была проведена в полном объеме.

Итоговая аттестационная работа оценивается аттестационной комиссией. Пояснительная записка к ИАР является обязательным документом.

Выполнение итоговой аттестационной работы является обязательной.

### **Требования к пояснительной записке к ИАР**

1. Тема итоговой аттестационной работы.
2. Введение.
3. Актуальность, цель и задачи работы.
4. Стек используемых технологий.
5. Описание ход работы над проектом.
6. Анализ результатов работы.
7. Заключение.

По результатам выполнения ИАР аттестационная комиссия принимает решение о присвоении слушателям по результатам освоения дополнительной профессиональной программы профессиональной переподготовки квалификации «Специалист по созданию и сопровождению информационных ресурсов», о предоставлении права заниматься профессиональной деятельностью в сфере проектирования, разработки и интеграции информационных ресурсов в локальной сети и информационно-телекоммуникационной сети «Интернет» и выдаче диплома о профессиональной переподготовке.

### III. ОСНОВНОЕ СОДЕРЖАНИЕ ПРОГРАММЫ

#### 3.1. План учебной деятельности

Результаты обучения	Учебные действия/ формы текущего контроля	Используемые ресурсы/ инструменты/технологии
РО1. Применять выбранные языки программирования для написания программного кода	Лекции. Выполнение задания, разработка алгоритмов решений в соответствии с требованиями технического задания. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО2. Разрабатывает различные веб-архитектуры.	Лекции. Выполнение задания, включающего этапы создания программного кода в соответствии с техническим заданием. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО3. Применяет методы анализ и проверка исходного программного кода.	Лекции. Выполнение задания, включающего этапы анализа и проверки исходного программного кода. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО4. Применять методы и средства проектирования программного обеспечения.	Лекции. Выполнение задания, включающего использование типовых решений и шаблонов проектирования программного обеспечения. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции
РО5. Применяет системы управления проектами и задачами.	Лекции. Выполнение задания, включающего использование типовых решений при выявлении требований к программному обеспечению. Тесты	Материалы электронного курса в системе электронного обучения СФУ «е-Курсы». Видеоконференции

#### 3.2. Виды и содержание самостоятельной работы

Самостоятельная работа слушателя (СРС) предполагает углубление и закрепление теоретических знаний. СРС включает следующие виды самостоятельной деятельности: самостоятельное углубленное изучение вопросов программы, выполнение индивидуальных заданий, подготовка к тестированию и приобретение опыта работы в рамках электронного курса. Выполнение СРС предполагается в дистанционном режиме в рамках электронного курса.

# РАБОЧАЯ ПРОГРАММА

## дисциплины (модуля)

### «Верстка веб-приложений»

#### 1. Аннотация

Дисциплина «Верстка веб-интерфейсов» знакомит слушателей с основами верстки веб-страниц без погружения в специализированные фреймворки. В рамках этого модуля предлагается изучить современные техники верстки, язык гипертекстовой разметки, как основу любой из техник, а также язык описания внешнего вида веб-страниц – CSS, который тоже является неотъемлемой частью любого современного веб-интерфейса. В результате обучения слушатели будут обладать умениями достаточными для верстки одностраничного веб-сайта (лендинга), а именно, для создания веб-страницы на языке HTML, размещения элементов интерфейса в соответствии с требованиями дизайн-проекта, создания стилей оформления внешнего вида веб-страницы на языке CSS.

#### Цель дисциплины (результаты обучения)

PO1. Применяет языки программирования для решения профессиональных задач.

PO2. Разрабатывает различные веб-архитектуры.

PO3. Применяет методы анализ и проверка исходного программного кода.

#### 2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
<b>Модуль 1. Верстка веб-интерфейсов (60 часов)</b>			
1.1. Введение в язык гипертекстовой разметки (HTML) (6 ч.)	Базовые основы HTML. Структура документа. Понятие и виды верстки веб-приложений (1 ч.)	Структура HTML-документа и группы тегов (2 ч.). <i>Задание 1.</i> Создание первой HTML-страницы	Изучение HTML-тегов и структуры HTML-документа. Изучение теоретических материалов. Тестирование (3 ч.)
1.2. Введение в CSS (6 ч.)	Основы CSS, назначение. Приоритет стилей. Стиливые свойства и их значения. Синтаксис CSS. Способы добавления стилей на страницу (1 ч.)	Использование CSS-стилей. Приоритет стилей (2 ч.). <i>Задание 2</i> Создание CSS-стилей для оформления интерфейса веб-приложения	Синтаксис и свойства CSS. Изучение теоретических материалов. Тестирование (3 ч.)



№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
1.3. CSS-селекторы (12 ч.)	Понятие селектора. Виды селекторов. Создание классов и идентификаторов. Группирование стилей. Иерархия селекторов. Переопределение стилей (2 ч.)	Виды CSS-селекторов. Группирование стилей (4 ч.). <i>Задание 3.</i> Разделение HTML-кода и стилевого оформления интерфейса веб-приложения	Классы, идентификаторы и контекстные селекторы. Изучение теоретических материалов. Тестирование (6 ч.)
1.4. Блочная верстка (12 ч.)	Введение в верстку, назначение. Виды верстки. Верстка HTML-таблицами. Пример табличной верстки по заданному макету. Понятие и преимущества блочной верстки. Тег DIV. Блочная модель. Пример блочной верстки по заданному макету (2 ч.)	Блоки DIV и позиционирование (4 ч.). <i>Задание 4.</i> Блочная верстка страницы по собственному макету	Табличная верстка. Достоинства и недостатки таблиц. Изучение теоретических материалов. Тестирование (6 ч.)
1.5. At-правила в CSS (12 ч.)	Понятие at-правил в CSS, назначение. Синтаксис at-правил. Размер экрана устройства и правило @media (2 ч.)	Виды at-правил и директива @media (4 ч.). <i>Задание 5.</i> Проектирование CSS-стилей для разных типов устройств	At-правила. Изучение теоретических материалов. Тестирование (6 ч.)
1.6. Адаптивная верстка (12 ч.)	Понятие и преимущества адаптивной верстки. Пример адаптивной верстки по заданному макету (2 ч.)	Динамическая адаптация интерфейса под размеры экрана устройства (4 ч.). <i>Задание 6.</i> Адаптивная верстка страницы по собственному макету	Изучение принципов адаптивной верстки. Изучение теоретических материалов. Тестирование (6 ч.)

### **3. Условия реализации программы дисциплины**

#### **Организационно-педагогические условия реализации программы**

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

#### **Материально-технические условия реализации программы**

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, текстовый редактор.

#### **Учебно-методическое и информационное обеспечение программы**

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

#### **Содержание комплекта учебно-методических материалов**

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

#### **Литература**

##### *Основная литература*

1. Шафер С., Сергеев А.П. HTML, XHTML, и CSS. Библия пользователя. – М.: Диалектика, 2011.
2. Лоусон Б., Шарп Р. Изучаем HTML 5. Библиотека специалиста. 2-е издание. – СПб.: Питер, 2012.

*Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины*

1. Самоучитель HTML4 [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/samhtml>.

2. Верстка сайта на HTML [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/html5>.

3. Самоучитель CSS [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/samcss>.

Учебник CSS3 [Электронный ресурс]. – Режим доступа: [https://professorweb.ru/my/css/css\\_theory/level1/css\\_index.php](https://professorweb.ru/my/css/css_theory/level1/css_index.php).

#### **4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)**

**Форма аттестации по дисциплине** — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100 от общего прогресса по курсу.

#### **Примеры тестовых заданий**

*Пример тестового задания по типу «Множественный выбор»*

1. Какой HTML-код применяется для подключения внешнего CSS-файла?
  - a) `<style>mystyle.css</style>`
  - b) `<style>@mystyle.css</style>`
  - c) `<link rel=«stylesheet» href=«mystyle.css»>`
  - d) `<link>@import url(mystyle.css)</link>`
  - e) `<stylesheet>mystyle.css</stylesheet>`

#### **Типовое практическое задание**

##### **Создание CSS-стилей для оформления интерфейса веб-приложения**

Создайте и отформатируйте контент веб-страницы в соответствии с перечисленными требованиями:

1. Создайте HTML-страницу с помощью текстового редактора. Рекомендуется использовать любой текстовый редактор, который поддерживает подсветку синтаксиса, работу с вкладками и проверку синтаксических ошибок.
2. В контейнере `<body>` добавьте параграф `<p>`. Внутри параграфа поместите строчный тег `<span>`, в который введите текст лозунга: наш долг – учиться отлично.
3. Создайте css-файл `style.css` и подключите его к html-документу.
4. В файле `style.css` опишите следующие стили оформления:
  - 4.1. Уберите все отступ на странице.

- 4.2. Задайте контейнеру <p> класс slogan.
- 4.3. В классе slogan задайте:
  - 4.3.1. цвет фона – #ED1C24,
  - 4.3.2. минимальную высоту блока параграфа – 100px,
  - 4.3.3. параметры шрифта: размер 60px, шрифт – Impact, sans-serif,
  - 4.3.4. перенос текста – nowrap,
  - 4.3.5. позиционирование контейнера – относительное,
  - 4.3.6. горизонтальное положение текста – по центру
- 4.4. Для строчного тега <span> задайте следующие стили:
  - 4.4.1. абсолютное позиционирование,
  - 4.4.2. отступ слева – 0,
  - 4.4.3. отступ справа – 0,
  - 4.4.4. отступ снизу – -13px.
5. Результат должен выглядеть, как представлено на рисунке.



Рисунок – Результат выполнения задания

### Масштабируемая верстка страницы

Сверстайте веб-страницу по образцу, используя блочную масштабируемую верстку.

Условия:

1. Предполагается, что содержимое страницы будет растягиваться и сжиматься при изменении ширины окна браузера.
2. В растянутом состоянии страница выглядит как на рисунке 1.

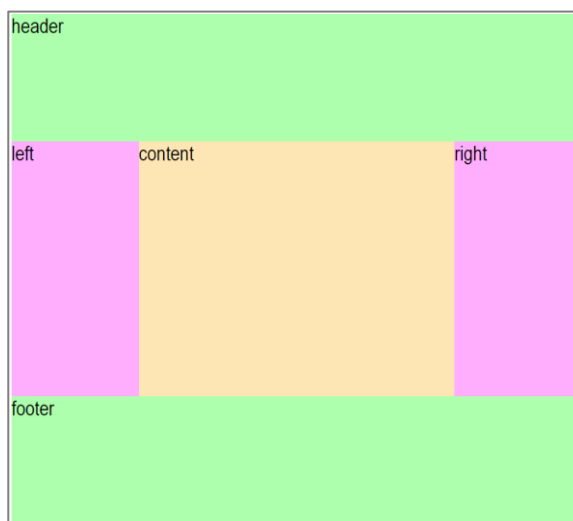


Рисунок 1 – Страница в растянутом состоянии

3. В сжатом состоянии страница выглядит как на рисунке 2.

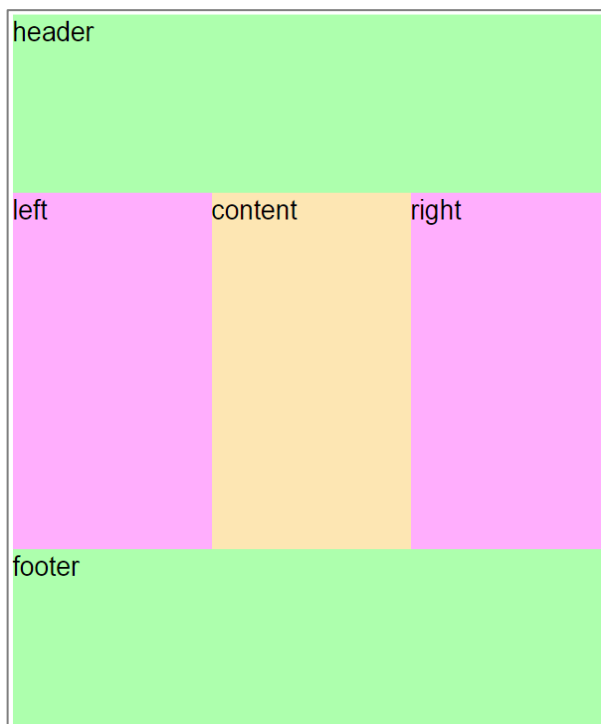


Рисунок 2 – Страница в сжатом состоянии

### Последовательность работы и рекомендации

1. Создайте HTML-страницу со стандартной структурой с помощью текстового редактора. Рекомендуется использовать любой текстовый редактор, который поддерживает подсветку синтаксиса, работу с вкладками и проверку синтаксических ошибок.

2. В блоке `<body>` создайте блок `<div>` с классом `container`. Внутри блока `<div>` поместите еще пять дочерних блоков `<div>`, с классами `header`, `left`, `right`, `content` и `footer`.

3. Создайте `css`-файл `style.css` и подключите его к `html`-документу.

4. В файле `style.css` опишите стили оформления блоков по следующим рекомендациям:

4.1. Для блока `container` укажите ширину 70%, минимальную ширину 600px, максимальную ширину 1200px. Можно задать меньше значения в зависимости от разрешения Вашего экрана.

4.2. Для левого и правого блоков задайте одинаковую высоту и ширину в px. В таком случае эти блоки останутся не «резиновыми», т.е. при растягивании или сжатии окна размер этих блоков будет статичным.

4.3. Для блока `content` укажите высоту равную высоте блоков `left` и `right`. С помощью свойства `margin` укажите отступы слева и справа равные ширине блоков `left` и `right` соответственно.

4.4. Для левого блока укажите свойство `float: left`, а для правого блока – `float: right`. Это позволит выстроить блоки `left`, `content` и `right` в одну линию, а не друг под другом.

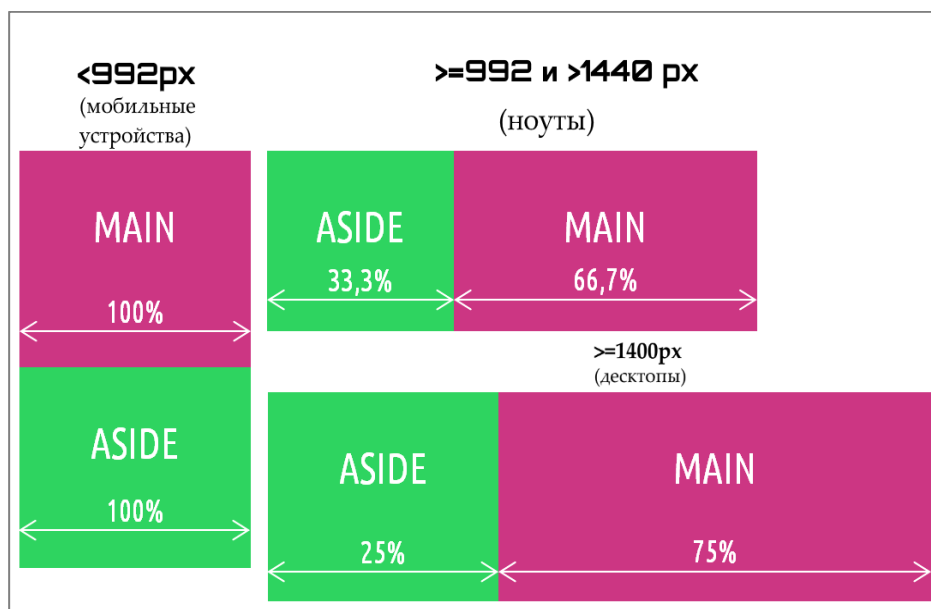
## Адаптивная верстка страницы

Сверстайте веб-страницу по образцу и заданным размерам, используя адаптивную верстку

Условия:

1. Предполагается, что страница будет открываться на мобильных устройствах, ноутбуках и на мониторах (десктопах).

2. Используйте размеры, указанные на изображении:



### Последовательность работы и рекомендации

1. Создайте HTML-страницу со стандартной структурой с помощью текстового редактора. Рекомендуется использовать любой текстовый редактор, который поддерживает подсветку синтаксиса, работу с вкладками и проверку синтаксических ошибок.

2. В блоке `<body>` создайте блок `<div>` с классом `container`. Внутри блока `<div>` поместите еще два дочерних блока `<div>`, с классами `main` и `aside`.

3. Создайте `css`-файл `style.css` и подключите его к `html`-документу.

4. В файле `style.css` опишите следующие стили оформления:

4.1. Для блока `body` уберите внешние отступы.

4.2. Для блока `container`:

4.2.1. Сделайте все дочерние элементы «резиновыми» (свойство `display`).

4.2.2. Укажите свойство `flex-wrap`, чтобы разрешить гибким блокам располагаться в несколько строк.

4.3. Для элементов `aside` и `main` укажите:

4.3.1. ширину в 100%,

4.3.2. высоту – 100px,

4.3.3. цвет фона для `aside` – `#ffc5f8`,

4.3.4. цвет фона для `main` – `#fdeea6`.

4.4. Сформируйте размеры блоков `aside` и `main` для минимальной ширины экрана 990px. Для этого создайте первый медиа-запрос, в котором укажите следующие свойства:

4.4.1. минимальная ширина элемента 990px,

4.4.2. свойство `flex` для `aside` – `flex: 33.3%` или `<=990px: 33.3%`

4.4.3. свойство `flex` для `main` – `flex: 66.6%`,

4.4.4. укажите порядок, в котором будут представлены блоки при отображении на мобильных устройствах, как показано на рисунке: вверху `main`, внизу `aside`. Используйте свойство `order`.

4.5. Сформируйте размеры блоков `aside` и `main` для минимальной ширины экрана 1400px. Для этого создайте второй медиа-запрос укажите следующие свойства:

4.5.1. минимальная ширина элемента 1400px,

4.5.2. свойства `flex` и `max-width` для `aside` – 25%,

4.5.3. свойства `flex` и `max-width` для `main` – 75%.

### Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

**РАБОЧАЯ ПРОГРАММА**  
**дисциплины (модуля)**  
**«Основы языка программирования JavaScript для создания**  
**интерактивных веб-страниц»**

**1. Аннотация**

Данный модуль направлен на изучение основ языка программирования JavaScript, который является ключевым инструментом для создания интерактивных и динамических веб-страниц. Модуль охватывает базовые концепции и синтаксис JavaScript, включая переменные, типы данных, операторы, функции, объекты, массивы и события. Основное внимание уделяется практическому применению JavaScript в веб-разработке: работа с DOM (Document Object Model).

В рамках курса студенты будут выполнять практические задания и проекты, направленные на закрепление теоретических знаний и развитие навыков программирования. По завершении модуля слушатели смогут самостоятельно создавать интерактивные веб-страницы, улучшая пользовательский интерфейс и опыт взаимодействия с веб-сайтами.

Основная цель модуля: изучить основные концепции и синтаксис JavaScript.

**Цель дисциплины (результаты обучения)**

По окончании обучения на данной дисциплине слушатели будут способны:

PO1. Применяет языки программирования для решения профессиональных задач.

PO2. Разрабатывает различные веб-архитектуры.

PO3. Применяет методы анализ и проверка исходного программного кода.

PO4. Применять методы и средства проектирования программного обеспечения.

**2. Содержание**

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
<b>Модуль 2. Основы языка программирования JavaScript для создания интерактивных веб-страниц (96 часов)</b>			
2.1. Основы JavaScript (12 ч.)	Скриптовый язык. Процесс выполнения (обработки) кода (2 ч.)	Создание веб-страницы с использованием JavaScript (4 ч.)	История и особенности JavaScript (6 ч.)
2.2. Типы данных (12 ч.)	Основные “примитивные” типы данных и операции с ними	Использование примитивных типов данных в JavaScript (4 ч.)	Примитивные типы данных (Number, String, Boolean, null, undefined, Symbol,



№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
	(2 ч.)		BigInt). Операции с примитивными типами (6 ч.)
2.3. Преобразование типов (12 ч.)	Явное преобразование типов. Неявное преобразование типов (2 ч.)	Явное и неявное преобразование типов в JavaScript (4 ч.)	Тонкости с преобразованием (6 ч.)
2.4. Выражение и операторы (12 ч.)	Что такое выражения. Что такое операторы. Циклы и условия. const, let, var. (2 ч.)	Использование выражений и операторов для решения задач (4 ч.)	Арифметические операторы. Сравнительные операторы. Логические операторы. Условные операторы. Операторы присваивания (6 ч.)
2.5. Функции (12 ч.)	Виды функций. Типы функций. Рекурсия. (2 ч.)	Создание и использование функций в JavaScript (4 ч.)	Функциональное программирование (6 ч.)
2.6. Объекты (12 ч.)	Способы создания объектов. Что такое объект. Итерируемые объекты. Копирование объектов. Ключевое слово «this» (2 ч.)	Создание и работа с объектами в JavaScript (4 ч.)	Создание объектов и доступ к их свойствам. Методы объектов (6 ч.)
2.7. Массивы (12 ч.)	Методы для работы с массивами (2 ч.)	Манипуляции с массивами в JavaScript (4 ч.)	Методы поиска и сортировки массивов (6 ч.)
2.8. DOM (12 ч.)	DOM. Поиск и навигация по элементам. Контент и атрибуты. (2 ч.)	Манипулирование DOM с помощью JavaScript (4 ч.)	Основные понятия DOM. Поиск элементов в DOM (getElementById, getElementsByName, querySelector) (6 ч.)

### 3. Условия реализации программы дисциплины

#### Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов,

презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

### **Материально-технические условия реализации программы**

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, текстовый редактор, Microsoft PowerPoint.

## **Учебно-методическое и информационное обеспечение программы**

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

### **Содержание комплекта учебно-методических материалов**

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

## **Литература**

### *Основная литература*

1. Дэвид Флэнаган, JavaScript. Полное руководство. Седьмое издание, 720 с, 2021.
2. Марейн Хавербеке, Выразительный JavaScript. Современное веб-программирование. Третье издание, 480 с, 2019.
3. Эрик Фримен, Элизабет Робсон, Изучаем программирование на JavaScript, 510 с, 2022.

## **4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)**

### **Форма аттестации по дисциплине — зачет.**

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100 от общего прогресса по курсу.

## **Примеры тестов для контроля знаний**

### *Пример тестового задания по типу «Выбор верного ответа»*

#### **Вопрос:**

Что такое поднятие (hoisting) в JavaScript и как оно влияет на переменные, объявленные с помощью ключевого слова var?

#### **Варианты ответа:**

а) Поднятие — это процесс, при котором объявления переменных и функций переносятся в начало их области видимости, но инициализации остаются на месте, что означает, что переменные, объявленные с помощью var,

могут быть доступны до их фактического объявления, но их значение будет undefined.

б) Поднятие — это процесс, при котором объявления переменных и функций переносятся в начало их области видимости, вместе с их инициализацией, что означает, что переменные, объявленные с помощью var, всегда будут иметь свои значения, даже если они используются до объявления.

с) Поднятие — это процесс, который применяется только к функциям, а не к переменным, что означает, что переменные, объявленные с помощью var, не поднимаются и не могут быть использованы до их фактического объявления.

#### **Правильный ответ:**

а) Поднятие — это процесс, при котором объявления переменных и функций переносятся в начало их области видимости, но инициализации остаются на месте, что означает, что переменные, объявленные с помощью var, могут быть доступны до их фактического объявления, но их значение будет undefined

## **Типовые практические задания**

### **Тема «Основы JavaScript»**

**Задание:** Создание первой веб-страницы с использованием JavaScript

**Цель:** научиться создавать и подключать JavaScript к HTML-документу, а также взаимодействовать с элементами страницы.

Описание задания:

Вам предстоит создать простую веб-страницу, используя HTML и JavaScript. На этой странице будет приветственное сообщение, которое появится при загрузке страницы, и кнопка, по нажатию на которую будет выводиться дополнительное сообщение.

#### **Шаги выполнения задания:**

1 Создайте HTML-документ:

Создайте файл с именем index.html.

Определите структуру HTML-документа, включая DOCTYPE, <html>, <head>, и <body> теги.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First JavaScript Page</title>
</head>
<body>
  <h1>Welcome to My Web Page</h1>
  <p id="welcome-message">Hello, World!</p>
  <button id="my-button">Click Me!</button>
  <p id="additional-message"></p>
  <script src="script.js"></script>
```

```
</body>  
</html>
```

## 2. Создайте JavaScript-файл:

Создайте файл с именем script.js.

Подключите этот файл к вашему HTML-документу с помощью тега `<script>` (уже сделано в шаге 1).

## 3. Напишите JavaScript код:

В файле script.js напишите код, который будет выводить приветственное сообщение в консоль при загрузке страницы.

Добавьте функциональность для кнопки, чтобы при ее нажатии на странице отображалось дополнительное сообщение.

```
// Вывод приветственного сообщения в консоль при загрузке страницы  
console.log("Welcome to My First JavaScript Page!");
```

```
// Получаем элементы по их ID  
const button = document.getElementById('my-button');  
const additionalMessage = document.getElementById('additional-message');
```

```
// Добавляем обработчик события на кнопку  
button.addEventListener('click', function() {  
    additionalMessage.textContent = "You clicked the button!";  
});
```

## 4. Проверьте работу страницы:

Откройте index.html в вашем браузере.

Убедитесь, что приветственное сообщение отображается в консоли разработчика.

Нажмите на кнопку и убедитесь, что дополнительное сообщение появляется на странице.

### **Критерии оценки:**

HTML-документ корректно структурирован и содержит необходимые элементы.

JavaScript-файл успешно подключен к HTML-документу.

При загрузке страницы в консоли выводится приветственное сообщение.

При нажатии на кнопку на странице появляется дополнительное сообщение.

Код чистый и хорошо организованный.

### **Рекомендации:**

Используйте понятные и описательные имена для переменных и функций.

Протестируйте вашу страницу в нескольких браузерах для проверки кросс-браузерной совместимости.

Используйте комментарии в коде для пояснения работы отдельных частей.

## **Заключение:**

После выполнения этого задания у вас будет базовое представление о создании и подключении JavaScript к веб-странице, а также об обработке событий на странице. Это первый шаг на пути к более сложным и интерактивным веб-приложениям.

## **Тема «Primitive Value»**

**Задание:** Использование примитивных типов данных в JavaScript

**Цель:** научиться работать с примитивными типами данных в JavaScript, используя их в различных операциях и выражениях.

### **Описание задания:**

Вам предстоит создать скрипт, который продемонстрирует использование различных примитивных типов данных в JavaScript. Вы будете работать с типами данных: Number, String, Boolean, null, undefined, Symbol и BigInt. Также вы научитесь использовать основные операции и методы, применимые к этим типам данных.

Шаги выполнения задания:

1. Создайте HTML-документ:

Создайте файл с именем index.html.

Определите структуру HTML-документа, включая DOCTYPE, <html>, <head>, и <body> теги.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Primitive Data Types in JavaScript</title>
</head>
<body>
  <h1>Primitive Data Types in JavaScript</h1>
  <div id="output"></div>

  <script src="script.js"></script>
</body>
</html>
```

2. Создайте JavaScript-файл:

Создайте файл с именем script.js.

Подключите этот файл к вашему HTML-документу с помощью тега <script> (уже сделано в шаге 1).

3. Напишите JavaScript код:

В файле script.js создайте переменные для каждого примитивного типа данных и продемонстрируйте их использование.

Выведите результаты в консоль и на веб-страницу.

```

// Примитивные типы данных

// Number
const number = 42;
console.log("Number:", number);

// String
const string = "Hello, World!";
console.log("String:", string);

// Boolean
const boolean = true;
console.log("Boolean:", boolean);

// Null
const nullValue = null;
console.log("Null:", nullValue);

// Undefined
let undefinedValue;
console.log("Undefined:", undefinedValue);

// Symbol
const symbol = Symbol('symbol');
console.log("Symbol:", symbol);

// BigInt
const bigInt = BigInt(1234567890123456789012345678901234567890n);
console.log("BigInt:", bigInt);

// Вывод на веб-страницу
const output = document.getElementById('output');
output.innerHTML = `
  <p>Number: ${number}</p>
  <p>String: ${string}</p>
  <p>Boolean: ${boolean}</p>
  <p>Null: ${nullValue}</p>
  <p>Undefined: ${undefinedValue}</p>
  <p>Symbol: ${symbol.toString()}</p>
  <p>BigInt: ${bigInt}</p>
`;

```

#### 4. Проверьте работу страницы:

Откройте `index.html` в вашем браузере.

Убедитесь, что все примитивные типы данных правильно отображаются в консоли и на веб-странице.

#### **Критерии оценки:**

HTML-документ корректно структурирован и содержит необходимые элементы.

JavaScript-файл успешно подключен к HTML-документу.

Все примитивные типы данных корректно объявлены и используются.

Результаты работы скрипта выводятся в консоль и на веб-страницу.

Код чистый и хорошо организованный.

**Рекомендации:**

Используйте понятные и описательные имена для переменных.

Протестируйте вашу страницу в нескольких браузерах для проверки кросс-браузерной совместимости.

Используйте комментарии в коде для пояснения работы отдельных частей.

**Заключение:**

После выполнения этого задания у вас будет хорошее понимание примитивных типов данных в JavaScript и их использования в различных операциях и выражениях. Это знание является основой для дальнейшего изучения и эффективного использования JavaScript.



### Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла	4 балла	5 баллов
Критерий	Реализован только вывод поля и персонажа	Реализовано только перемещение и выполнены предыдущие требования	Реализован поворот персонажа по направлению движения и выполнены предыдущие требования	Реализовано плавное перемещение персонажа и выполнены предыдущие требования	Реализован связанный с PlayerController класс BluePrint и выполнены предыдущие требования.

**РАБОЧАЯ ПРОГРАММА**  
**дисциплины (модуля)**  
**«Разработка интерактивных веб-интерфейсов»**

**1. Аннотация**

«Разработка интерактивных веб-интерфейсов» рассматривается как продолжение развития навыков frontend-разработки. В рамках модуля предлагается изучение фреймворка Vue.js, основанного на языке JavaScript, и его возможностей по реализации функционала пользовательского интерфейса и логики взаимодействия пользователя с веб-страницей. В результате обучения слушатели будут обладать следующими умениями:

- создавать интерактивные веб-страницы на фреймворке Vue.js, которые могут взаимодействовать с пользователем;
- динамически изменять содержимое веб-страниц;
- работать с событиями, т.е. создавать функции, которые выполняют свой код только после того, как совершатся определенные действия, например, движение мыши или наведение указателя мыши на какой-либо HTML-элемент или нажатие на кнопку, ссылку, блочный элемент и т.д.;
- выполнять функции через заданные промежутки времени;
- изменять внешний вид веб-страницы или ее отдельных элементов путем автоматической подмены CSS-стилей (подсветка полей ввода, плавные переходы стилей оформления и пр.);
- Создавать анимированные сценарии с использованием CSS-анимации и возможностей Vue.js.

**Цель дисциплины (результаты обучения)**

По окончании обучения на данной дисциплине слушатели будут способны:

- PO1. Применяет языки программирования для решения профессиональных задач.
- PO2. Разрабатывает различные веб-архитектуры.
- PO3. Применяет методы анализ и проверка исходного программного кода.
- PO4. Применять методы и средства проектирования программного обеспечения.

## 2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
<b>Модуль 3. Разработка интерактивных веб-интерфейсов (60 часов)</b>			
3.1. JavaScript фреймворк Vue.js (6 ч.)	Введение в Vue.js. Синтаксис фреймворка. Взаимодействие с HTML. Понятие объекта в JS. JSON-формат описания объекта. Среда разработки. (1 ч.)	Выбор темы для создания лендинга. Подключение фреймворка Vue.js и создание первого проекта (2 ч.). <i>Задание 1.</i> Создание приложения “Hello, Vue!”	Основы языка JavaScript: переменные, объекты. Изучение теоретических материалов. Тестирование (3 ч.)
3.2. Шаблоны (6 ч.)	Понятие шаблона. Синтаксис шаблонов. Стандартные директивы. Пример создания шаблона. (1 ч.)	Взаимодействие Vue.js с объектной моделью документа HTML (2 ч.). <i>Задание 2.</i> Создание лендинга: укрупненная верстка страницы с использованием шаблонов для динамического формирования HTML-контента	JSON-формат описания объекта. Объектная модель документа (DOM). Изучение теоретических материалов. Тестирование (3 ч.)
3.3. События (12 ч.)	Виды событий. Подписка на события. Обработчики событий. Примеры обработки событий. (2 ч.)	События в Vue.js (4 ч.). <i>Задание 3.</i> Определение активных элементов лендинга и их привязка к событиям	Пользовательские события. Изучение теоретических материалов. Тестирование (6 ч.)
3.4. Компоненты (12 ч.)	Понятие компонента. Регистрация компонентов. Импорт компонентов. Пример создания и использования компонента (2 ч.)	Повторное использование шаблонов (4 ч.). <i>Задание 4.</i> Оформление фрагмента лендинга в отдельный компонент	Повторное использование компонентов. Изучение теоретических материалов. Тестирование (6 ч.)
3.5. Взаимодействие с CSS (12 ч.)	Связывание CSS-классов. Динамическое изменение CSS-классов стилей. Примеры взаимодействия с CSS (2 ч.)	Стилевое оформление лендинга. (4 ч.). <i>Задание 5.</i> Определение концепции стилового оформления лендинга. Создание файла CSS-стилей	Директива условия v-if и автоматическое изменение стилей элементов. Изучение теоретических материалов. Тестирование (6 ч.)
3.6. Анимация (12 ч/)	Анимация в Vue.js – общие сведения. Классы переходов.	Разработка анимационных сценариев (4 ч.).	Функции трансформирования,

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
	CSS-переходы. CSS-анимация. Примеры создания анимации (2 ч.)	<i>Задание б.</i> Изменение внешнего вида и поведения выбранного элемента(ов) с помощью нескольких ключевых кадров	переходов и анимации в CSS. Изучение теоретических материалов. Тестирование (6 ч.)

### **3. Условия реализации программы дисциплины**

#### **Организационно-педагогические условия реализации программы**

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

#### **Материально-технические условия реализации программы**

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Microsoft Visual Studio Code, текстовый редактор Atom (лицензия MIT).

#### **Учебно-методическое и информационное обеспечение программы**

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

#### **Содержание комплекта учебно-методических материалов**

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические

и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

## Литература

### *Основная литература*

1. Гаевский А.Ю., Романовский В.А. Самоучитель по созданию Web-страниц: HTML, JavaScript и Dynamic HTML. – Киев: А.С.К., 2002.
2. Фримен Э., Робсон Э., Матвеев Е. Изучаем программирование на JavaScript. – СПб.: Питер, 2015.

### *Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины*

1. Введение в фреймворк VueJS [Электронный ресурс]. – Режим доступа: <https://ru.vuejs.org/v2/guide/>.
2. Учебник JavaScript с интерактивными примерами [Электронный ресурс]. – Режим доступа: <https://www.wisdomweb.ru/JS/javascript-first.php>.
3. Веб-технологии для разработчиков [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web>.
4. Онлайн-среда для отладки и тестирования скриптов на VueJS [Электронный ресурс]. – Режим доступа: <https://scrimba.com/scrim/cQ3QVcr?pl=pXKqta>.

## **4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)**

### **Форма аттестации по дисциплине — зачет.**

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100 от общего прогресса по курсу.

### **Примеры тестовых заданий для контроля знаний**

#### *Пример тестового задания по типу «Множественный выбор»*

1. Какая директива фреймворка VueJS используется для отрисовки списка элементов массива в HTML-шаблонах?
  - a) v-for;
  - b) v-foreach;
  - c) v-while;
  - d) v-loop;

## Типовое практическое задание

### Создание приложения “Hello, Vue!”

Создайте свое первое приложение на языке JavaScript с использованием фреймворка Vue.js. Для этого придерживайтесь следующего порядка действий:

1. Создайте HTML-страницу с помощью текстового редактора. Рекомендуется использовать любой текстовый редактор, который поддерживает подсветку синтаксиса, работу с вкладками и проверку синтаксических ошибок.

1.1. В контейнере `<body>` добавьте контейнер `<div>` идентификатором `app`.

1.2. Внутри контейнер `<div>` поместите параграф `<p>` или строчный тег `<span>` (на Ваше усмотрение), в котором будет выводиться текст приветствия.

2. Создайте JS-файл `script.js`, в котором будет содержаться код Vue.js и подключите его к html-документу. Обратите внимание, что правильным местом подключения файла `script.js` будет самый конец контейнера `<body>`.

2.1. В файле скрипта объявите переменную `application`.

2.2. Свяжите эту переменную с элементом `<div>` по заданному идентификатору `app`.

2.3. В параметре `data` напишите переменную `message` со значением “Hello, Vue!”, которую выведете внутри уже созданного тега `<p>` или `<span>` в html-странице. Используйте для этого формат `mustache`, двойные фигурные скобки: `{{ message }}`.

3. Откройте созданную HTML-страницу в браузере, чтобы отобразить приветствие.

### Создание приложения «Поисковик»

Создайте приложение, в котором реализуйте мгновенный поиск каких-либо объектов из заданного списка, например, товаров или новостных публикаций.

Структура проекта `vueserach`:

```
vuesearch
|—js
  |—vue.js
|—css
  |—style.css
|—imgs
  |—small1.jpg
  |—small2.jpg
index.html
```

Приложение должно отвечать следующим требованиям (рисунок 1):

– Объекты для поиска должны быть представлены в виде списка.

- Каждая строка в списке должна включать миниатюру изображения (например, 30x30 пикселей) и текст краткого описания (10-15 слов). Поиск должен осуществляться только по словам из текста краткого описания.
- Все объекты списка должны быть представлены в виде массива объектов в модели Vue.
- Над списком требуется разместить текстовое поле поиска (input type=«text») для ввода символов, по которым будут фильтроваться объекты в списке.

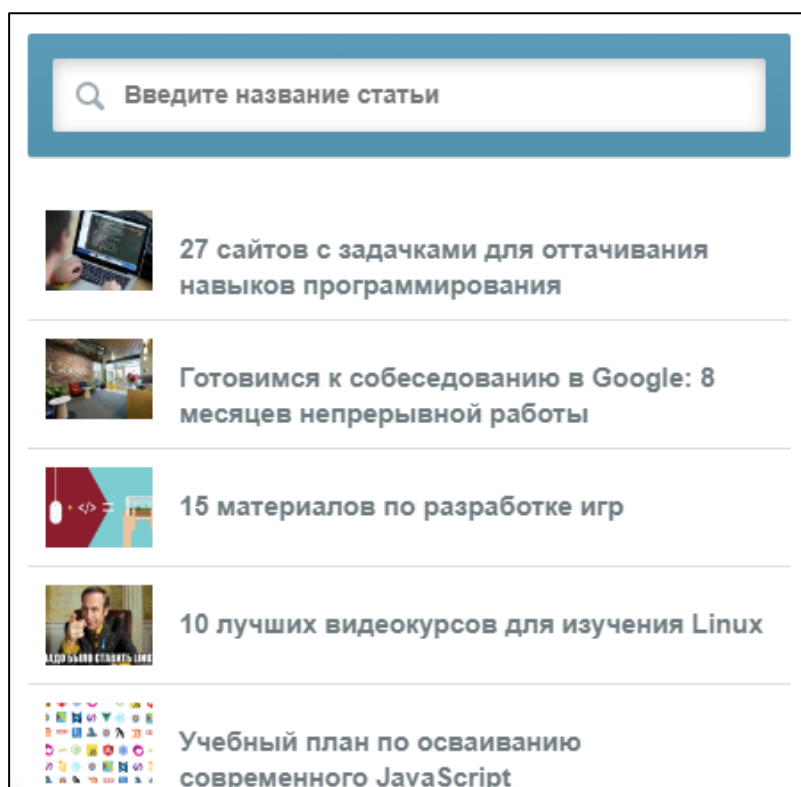


Рисунок 1 – Список объектов

В результате поиска и работы механизма фильтрации на странице должны быть отображены только объекты, удовлетворяющие введенным символам поиска (рисунок 2). В случае отсутствия совпадений – результирующий список должен быть пустым.

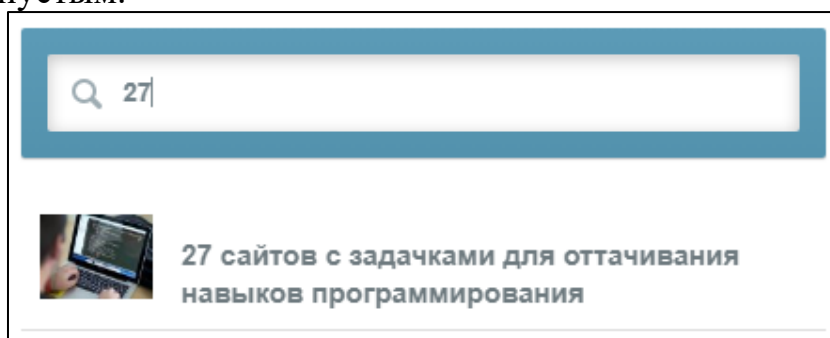


Рисунок 2 – Результат поиска и фильтрации

Используйте свободное стилевое оформление и верстку.

## Создание приложения «Переключаемая сетка»

Создайте приложение, в котором реализуйте разные режимы компоновки элементов

Структура проекта vuegride:

```
vuegride
|--js
    |--vue.js
|--css
    |--style.css
|--imgs
    |--large1.jpg
    |--large2.jpg
    |--small1.jpg
    |--small2.jpg
index.html
```

Приложение должно отвечать следующим требованиям (рисунок 1):

- В режиме компоновки элементов в виде сетки должна отображаться только картинка без текста. Размер картинки 280x280 px или другой.
- В режиме компоновки элементов в виде списка каждая строка в списке должна включать миниатюру изображения (например, 30x30 пикселей) и текст краткого описания (10-15 слов).
- Все отображаемые объекты должны быть представлены в виде массива объектов в модели Vue. Рекомендуемая структура для коллекции объектов «Статья»:

```
articles: [
  {
    «title»: «27 сайтов с задачками для оттачивания навыков
    программирования»,
    «url»: «#»,
    «image»: {
      «large»: «../imgs/large1.jpg»,
      «small»: «../imgs/small1.jpg»
    }
  },
  {
    «title»: «Готовимся к собеседованию в Google: 8 месяцев непрерывной
    работы»,
    «url»: «#»,
    «image»: {
      «large»: «../imgs/large1.jpg»,
      «small»: «../imgs/small1.jpg»
    }
  },
  ...
]
```



“title” – заголовок статьи,  
“url” – ссылка на сайт со статьей, в примере ссылка пустая – «#»,  
“image” – два варианта изображения: большое и маленькое. Изображения разместите в папке imgs.

– Нажав одну из кнопок в верхней панели, будет переключаться режим показа между сеткой, содержащей большие изображения (рисунок 3), и списком с меньшими изображениями и текстом (рисунок 4).



Рисунок 3 – Компоновка элементов в виде сетки (или плиток)

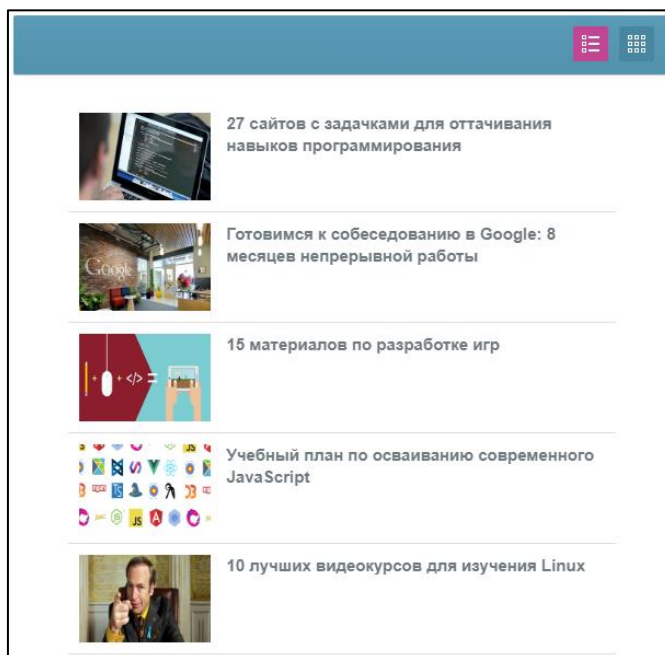


Рисунок 4 – Компоновка элементов в виде списка

Используйте свободное стилевое оформление и верстку.

## Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

# РАБОЧАЯ ПРОГРАММА СТАЖИРОВКИ

## 1. Аннотация

Основной задачей стажировки слушателей программы является закрепление в практической деятельности профессиональных компетенций, умений, навыков и знаний, полученных в ходе обучения, а также приобретение необходимых умений и практического опыта на конкретном рабочем месте.

**Цель стажировки** — приобретение слушателями программы практического опыта работы, а также освоение новых технологий, форм и методов организации труда непосредственно на рабочем месте.

### Планируемые результаты:

По окончании стажировки слушатели будут способны составлять формализованные описания решений и разрабатывать алгоритмы решений поставленных задач в соответствии с требованиями технического задания, дизайнерского документа или других принятых в организации нормативных документов; применять навыки работы с языком гипертекстовой разметки (HTML) в комбинации с CSS; осуществлять проверку работоспособности программного обеспечения на основе разработанных тестовых наборов данных; использовать при разработке программного обеспечения существующие типовые решения и шаблоны проектирования, будут способны применять методы анализа предметной области, выявления проблем, составлять ценностного предложения, проводить анализ рынка, выявлять конкурентные преимущества проекта, создавать презентации для представления проекта.

## 2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
<b>Стажировка (16 часов)</b>			
1. Общие вопросы (ознакомление с предприятием)		Ознакомление и изучение конкретной практической задачи (2 ч.)	
2. Практическая часть стажировки		Решение практической задачи (4 ч.) Интеграция собственного решения в общий проект (2 ч.)	
3. Подготовка отчетной документации			Составление отчета (4 ч.)

Содержание стажировки включает следующие этапы:

1. Ознакомление с нормативной базой, касающейся охраны труда и правил безопасной работы.

2. Знакомство с рабочим местом и должностными обязанностями, концептом общего тестового проекта.

3. Практическая деятельность, выполняемая под контролем руководителя стажировки. Обычно включает этапы:

3.1. Формирование отдельной практической задачи по общему проекту;

Содержание стажировки закрепляется индивидуальным планом прохождения стажировки (Приложение 1).

Продолжительность стажировки — 16 часов.

Стажировка носит индивидуальный или групповой характер и может предусматривать такие виды деятельности как:

- знакомство с предприятием, организационной структурой;
- изучение организации и технологии производства, работ;
- анализ производства;
- знакомство с общим проектом в сфере информационных технологий;
- работу с технической, нормативной и образовательной документацией;
- составление формализованных описаний решений поставленных задач;
- разработку алгоритмов решений поставленных задач в соответствии с требованиями технического задания;
- написание программного кода на языке гипертекстовой разметки (HTML) в комбинации с CSS;
- тестирование программного кода;
- интеграцию программных модулей;
- проведение анализа рынка и составление матрицы конкурентов;
- проведение анализа потребителей и составление ценностного предложения;
- формирование презентации проекта.

## Примеры типовых заданий стажировки

### 1. Верстка web-приложений

#### Задание по HTML – HTML разметка

Скачайте файл index.html и откройте его в редакторе и web-браузере. Найдите и исправьте в нем ошибки. Соблюдайте вложенность элементов, проверьте все атрибуты и кавычки, убедитесь, что элементы имеют закрывающиеся теги при необходимости.

#### Задание по HTML и CSS – Работа с текстом.

Отформатируйте текст по шаблону с использованием тегов HTML и стилизуйте его с применением CSS.

#### Задание по HTML и CSS – Абсолютное и относительное позиционирование.

Скачайте файлы задания и создайте композицию, которую вы видите на изображении, используя относительное и абсолютное позиционирование, а также z-index. После выполнения сравните с результатом.

#### Задание по HTML и CSS – Создание модального окна.

Скачайте файлы задания и создайте модальное окно, как на изображении, которое вы видите под заданием. После выполнения сравните с результатом.

### 2. Разработка интерактивных web-интерфейсов

#### Задание по HTML, CSS и Vue.js – Меню навигации.

Создайте меню навигации для сайта на основе перечисленных требований:

- Меню должно содержать не менее 4 вкладок: Главная, Проекты, Услуги, Контакты.
- Задайте каждой вкладке свой класс: для «Главной» – home, «Проекты» – projects, «Услуги» – services, «Контакты» – contacts.
- По умолчанию вкладка «Главная» должна быть активной.
- Активная вкладка должна отличаться по цвету от остальных вкладок.
- При клике на любую вкладку отобразите на странице название класса вкладки как подтверждение реактивного поведения.

#### Задание по HTML, CSS и Vue.js – «Хлебные крошки».

Создайте «хлебные крошки» для удобства навигации по сайту. Например, используйте такой формат: «Главная > Категория > Продукт» или «Главная / Категория / Продукт». Последний элемент должен быть активным, все остальные должны быть ссылками href и указывать на произвольные страницы.

### **Задание по HTML, CSS и Vue.js – Поисковик.**

Создайте приложение, в котором реализуйте мгновенный поиск каких-либо объектов из заданного списка, например, товаров или новостных публикаций. Приложение должно отвечать следующим требованиям:

- Объекты для поиска должны быть представлены в виде списка.
- Каждая строка в списке должна включать миниатюру изображения (например, 30x30 пикселей) и текст краткого описания (10–15 слов). Поиск должен осуществляться только по словам из текста краткого описания.
- Все объекты списка должны быть представлены в виде массива объектов в модели Vue.
- Над списком требуется разместить текстовое поле поиска (input type=«text») для ввода символов, по которым будут фильтроваться объекты в списке.
- В результате поиска и работы механизма фильтрации на странице должны быть отображены только объекты, удовлетворяющие введенным символам поиска. В случае отсутствия совпадений — результирующий список должен быть пустым.

### **3. Условия реализации программы стажировки**

#### **Организационные и педагогические условия реализации программы**

Обучение по программе стажировки реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Материал практических занятий представляется в виде синхронных занятий, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Стажировка проводится под руководством назначенного руководителя из числа профессорско-преподавательского состава Университета, а также руководителя из состава организации, структурных подразделениях организации, материально-техническое обеспечение которой соответствует профилю программы.

#### **Учебно-методическое и информационное обеспечение**

По данному модулю используется электронный УМК. УМК предполагает использование разных типов материалов, сопровождающих учебный процесс, включая информационные, обучающие и контролирующие. На платформе электронных курсов размещаются задания, приводится перечень необходимых для изучения материалов. Обучающиеся могут на протяжении прохождения стажировки обращаться к теоретической базе знаний.

#### **4. Оценка качества освоения программы стажировки (формы аттестации, оценочные и методические материалы)**

В качестве подтверждения прохождения стажировки на базе предприятий, организаций, учреждений, для зачета результатов обучения слушателями предъявляется дневник прохождения стажировки (Приложение 2) (*отчет в виде дневника прохождения практики*).

Программу составили:

Канд. техн. наук, доцент,  
доцент кафедры Информационных систем  
Института космических  
и информационных технологий СФУ

Е.А. Чжан

Канд. техн. наук, доцент,  
доцент кафедры Систем искусственного интеллекта  
Института космических  
и информационных технологий СФУ

Р.В. Брежнев

Руководитель программы:

Канд. техн. наук, доцент,  
доцент кафедры Информационных систем  
Института космических  
и информационных технологий СФУ

Е.А. Чжан

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

*Наименование образовательной организации*

**Индивидуальный план слушателя, направляемого на стажировку**

**Фамилия, имя, отчество** \_\_\_\_\_

**Место работы и должность/статус** \_\_\_\_\_

**Название предприятия (организации), где проводится стажировка**

\_\_\_\_\_

**Город** \_\_\_\_\_

**Цель стажировки** \_\_\_\_\_

\_\_\_\_\_

**Срок стажировки с «\_\_\_» \_\_\_\_\_ 2024 г. по «\_\_\_» \_\_\_\_\_ 2024 г.**

**Приказ по вузу от «\_\_\_» \_\_\_\_\_ 2024 г. № \_\_\_\_\_**

**План стажировки**

№ п.п.	Перечень разрабатываемых (изучаемых) вопросов, виды работ	Количество часов	Форма отчета
1.			Дневник стажировки
2.			
3.	Заполнение дневника стажировки		

СОГЛАСОВАНО

\_\_\_\_\_  
(должность ответственного)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(расшифровка подписи) лица,  
направляющего на стажировку)



**Наименование стажировочной площадки**

УТВЕРЖДАЮ  
 Руководитель стажировочной площадки  
 \_\_\_\_\_ ФИО  
 «\_\_\_\_\_» \_\_\_\_\_ 2024 г.  
 М.П.

**ДНЕВНИК  
 прохождения стажировки**

\_\_\_\_\_,  
 (фамилия, имя, отчество специалиста (стажера),  
 проходящего обучение в рамках дополнительной профессиональной программе  
 переподготовки «Разработка веб-приложений»

Цель стажировки:

\_\_\_\_\_

Руководители стажировки (от организации): \_\_\_\_\_  
 (должность) (ФИО)

**1. Дневник**

Дата	Выполняемая работа	Вопросы для консультантов и руководителей стажировки

**2. Краткий отчет о стажировке**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Дата

Подпись стажера

### 3. Заключение руководителя стажировки от принимающей организации

---

---

---

---

---

---

---

---

---

---

Руководитель стажировки

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(расшифровка подписи)

С заключением руководителя стажировки ознакомлен

\_\_\_\_\_

(подпись стажера)