

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кузбасский гуманитарно-педагогический институт

Факультет информатики, математики и экономики

УТВЕРЖДАЮ
Декан
А.В.Фомина
16 января 2025г.

Рабочая программа дисциплины

К.М.07.01.10 Программирование на JavaScript

Направление подготовки

Профессиональное обучение (по отраслям)

Направленность (профиль) подготовки

44.03.04 Компьютерный дизайн

Программа бакалавриата

Квалификация выпускника
бакалавр

Форма обучения
Очная, заочная

Год набора 2023

Новокузнецк 2025

Оглавление

1 Цель дисциплины	3
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.	4
3. Учебно-тематический план и содержание дисциплины.....	4
3.1 Учебно-тематический план	4
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.	6
5 Учебно-методическое обеспечение дисциплины.....	6
5.1 Учебная литература	7
5.2 Материально-техническое и программное обеспечение дисциплины.....	7
Программное обеспечение	7
5.2.1 Современные профессиональные базы данных и информационные справочные системы.....	8
6 Иные сведения и (или) материалы.....	8
6.1.Примерные темы письменных учебных работ	8
6.2. Примерные вопросы и задания / задачи для промежуточной аттестации	8

1 Цель дисциплины

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП): ПК-1.

Формируемые компетенции, индикаторы достижения компетенций, знания, умения, навыки

Таблица 1 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
ПК-1	ПК-1.3 Демонстрирует методы использования программных и аппаратных средств для создания объектов компьютерного дизайна.	Знать: – терминологию, используемую в клиент-серверных и веб-приложениях; – синтаксис языка JavaScript; – встроенные функции языка JavaScript; – модель взаимодействия скриптов и содержимого веб-страниц; – принципы объектно-ориентированного дизайна и проектирования с использованием JavaScript. Уметь: – формулировать требования к создаваемым программным комплексам; – формировать архитектуру Web-приложений для информатизации предприятий и организаций; – создавать клиентские скрипты на языке javascript; – тестировать и отлаживать клиентские скрипты. Владеть: – инструментальными средствами написания, отладки и тестирования JavaScript-кода; – фреймворками взаимодействия клиентских и серверных частей приложений; – фреймворками автоматизации тестирования приложений.

2 Объем и трудоемкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.

Таблица 2 – Объем и трудоемкость дисциплины по видам учебных занятий

Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах	Объем часов по формам обучения		
	ОФО	ЗФО	ОЗФО
1 Общая трудоемкость дисциплины	288	288	
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	120	37	
Аудиторная работа (всего):	84		
в том числе:			
лекции	28	8	
практические занятия, семинары			
практикумы			
лабораторные работы	56	20	
в интерактивной форме			
в электронной форме			
Внеаудиторная работа (всего):			
в том числе, индивидуальная работа обучающихся с преподавателем			
подготовка курсовой работы /контактная работа			
групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем)			
творческая работа (эссе)			
3 Самостоятельная работа обучающихся (всего)	168	251	
4 Промежуточная аттестация обучающегося:			
8 семестр - экзамен	36	9	

3. Учебно-тематический план и содержание дисциплины

3.1 Учебно-тематический план

Таблица 5 - Учебно-тематический план очной / заочной формы обучения

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоё мкость (всего час.)	Трудоемкость занятий (час.)						Формы ¹ текущего контроля и промежуточной аттестации успеваемости
			ОФО			ЗФО			
			Аудиторн. занятия		СРС	Аудиторн. занятия		СРС	
			лекц.	практ.		лекц.	практ.		
Семестр 8									
1. Введение в веб-разработку									
1.1	Основы веб-проектирования. Создание документов HTML.	20	2	4	14	2	2	16	ПР-6 – отчет по лабораторным работам ПР-4 реферат
1.2	Введение в технологии каскадных таблиц стилей.	20	2	6	12		2	18	ПР-6 – отчет по лабораторным работам

¹ УО - устный опрос, УО-1 - собеседование, УО-2 - коллоквиум, УО-3 - зачет, УО-4 – экзамен, ПР - письменная работа, ПР-1 - тест, ПР-2 - контрольная работа, ПР-3 эссе, ПР-4 - реферат, ПР-5 - курсовая работа, ПР-6 - научно-учебный отчет по практике, ПР-7 - отчет по НИРС, ИЗ –индивидуальное задание; ТС - контроль с применением технических средств, ТС-1 - компьютерное тестирование, ТС-2 - учебные задачи, ТС-3 - комплексные ситуационные задачи

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоём- кость (всего час.)	Трудоемкость занятий (час.)						Формы ¹ текущего контроля и промежуточной аттестации успеваемости
			ОФО			ЗФО			
			Аудиторн. занятия		СРС	Аудиторн. занятия		СРС	
лекц.	практ.	лекц.	практ.						
	Введение в веб- программирование на языке JavaScript								
2. Основы языка JavaScript									
2.1	Лексическая структура языка. Типы данных и значения.	20	2	4	14	2		18	ПР-6 – отчет по лабораторным работам
2.2	Переменные. Выражения и операторы.	20	2	4	14		2	18	ПР-6 – отчет по лабораторным работам
2.3	Инструкции. Объекты и массивы. JS Функции. JS Формы	20	2	4	14		2	18	ПР-6 – отчет по лабораторным работам
3. Клиентский JavaScript									
3.1	Документ и объекты страницы.	18	2	6	10	2		16	ПР-6 – отчет по лабораторным работам
3.2	Введение в браузерные события, общие свойства всех событий и приёмы работы с ними.	20	2	6	12		2	18	ПР-6 – отчет по лабораторным работам
3.3	Формы, элементы управления. Создание графических компонентов (виджетов).	18	2	6	10		2	16	ПР-6 – отчет по лабораторным работам
3.4	Cookies и механизм сохранения данных на стороне клиента	20	2	6	12		2	18	
3.5	Работа с протоколом HTTP.	20	4	4	12		2	18	
4. Серверный JavaScript									
4.1	Основы работы с платформой Node.js	18	2	2	14	2		16	ПР-6 – отчет по лабораторным работам
4.2	Модули Node. Фреймворк Express. Работа с серверами MondoDB и MySQL.	18	2	2	14		2	16	ПР-6 – отчет по лабораторным работам
4.3	Паттерн MVC. Загрузка файлов на сервер.	20	2	2	16		2	18	ПР-6 – отчет по лабораторным работам
	Промежуточная аттестация - экзамен	36						9	УО-4 – экзамен
ИТОГО по семестру 8									
	Всего:	288	28	56	168	8	20	251	

4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.

Для положительной оценки по результатам освоения дисциплины обучающемуся необходимо выполнить все установленные виды учебной работы. Оценка результатов работы обучающегося в баллах (по видам) приведена в таблице 4.

Таблица 4 - Балльно-рейтинговая оценка результатов учебной работы

8 Семестр

Составляющие	Сумма баллов	Учебная деятельность студента	Оценка в аттестации	Баллы (17 недель)
Текущая учебная деятельность	60	Посещение лекций (12 занятий).	1 балл (присутствие на лекции) 2 балла (активная работа, конспектирование)	12 - 24
		Лабораторные работы (18 работ).	1 балл (посещение занятия, выполнение работы на 51-65%) 2 балла (существенный вклад на занятии относительно всей группы, самостоятельность при выполнении работы, выполнение работы на 85,1-100%)	18 - 36
		Реферат	6 баллов (пороговое значение) 10 баллов (максимальное значение)	6 - 10
		Разработка проекта	15 баллов (пороговое значение) 30 баллов (максимальное значение)	15-30
Итого по текущей работе в семестре				51 - 100
Промежуточная аттестация (экзамен)	40	Теоретический вопрос 1.	5 баллов (пороговое значение) 10 баллов (максимальное значение)	5 - 10
		Теоретический вопрос 2.	5 баллов (пороговое значение) 5 баллов (максимальное значение)	5 - 10
		Защита проекта	10 баллов (пороговое значение) 20 баллов (максимальное значение)	10 - 20
Итого по промежуточной аттестации (экзамен)				20 – 40 б.
Суммарная оценка по дисциплине: Сумма баллов текущей и промежуточной аттестации				51 – 100 б.

Соотношение между оценками в баллах и их числовыми и буквенными эквивалентами устанавливается следующим образом:

Перевод баллов из 100-балльной шкалы в числовой и буквенный эквивалент

Сумма баллов для дисциплины	Оценка	Буквенный эквивалент
86 - 100	5	отлично
66 - 85	4	хорошо
51 - 65	3	удовлетворительно
0 - 50	2	неудовлетворительно

5 Учебно-методическое обеспечение дисциплины.

5.1 Учебная литература

Основная учебная литература

1. Брокшмидт, К. Пользовательский интерфейс приложений для Windows 8, созданных с использованием HTML, CSS и JavaScript : учебный курс / К. Брокшмидт. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 396 с. – URL: <http://biblioclub.ru/index.php?page=book&id=429247> (дата обращения: 26.12.2019).
2. Хоган, Б. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов = WebDevelopmentRecipes / Б. Хоган. - Санкт-Петербург [и др.] : Питер, 2013. - 288 с. - ISBN 978-5-459-01510-2. - Текст : непосредственный. (11 экз.)

Дополнительная учебная литература

1. Богданов, М.Р. Разработка клиентских приложений Web-сайтов: курс / М.Р. Богданов. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2010. – 228 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=233745> (дата обращения: 23.10.2020). – Текст : электронный.
2. Громов, Ю. Ю. Основы Web-инжиниринга: разработка клиентских приложений : учебное пособие / Ю. Ю. Громов, О. Г. Иванова, С. В. Данилкин ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». – Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2012. – 240 с. – URL: <http://biblioclub.ru/index.php?page=book&id=277648> (дата обращения: 26.12.2019). – Текст : электронный. <http://biblioclub.ru/index.php?page=book&id=277648>
3. Кингсли, Х. Э. JavaScript в примерах : учебное пособие / Х. Э. Кингсли, Х. К. Кингсли. — Москва : ДМК Пресс, 2009. — 272 с. — ISBN 978-5-94074-668-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/1271> (дата обращения: 24.10.2020). — Режим доступа: для авториз. пользователей.
4. Хэррон, Д. Node.js. Разработка серверных веб-приложений в JavaScript / Д. Хэррон ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2012. — 144 с. — ISBN 978-5-94074-809-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/50571> (дата обращения: 24.10.2020). — Режим доступа: для авториз. пользователей.
5. Штефен, В. Разработка приложений для Windows 8 с помощью HTML5 и JavaScript. Подробное руководство : руководство / В. Штефен ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2013. — 344 с. — ISBN 978-5-94074-921-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/58696> (дата обращения: 24.10.2020). — Режим доступа: для авториз. пользователей.

5.2 Материально-техническое и программное обеспечение дисциплины.

Программное обеспечение

В обучении используются информационные технологии на базе компьютерных классов учебного корпуса №4 (пр. Металлургов 19):

- лекционные занятия ведутся с использованием презентаций и программного обеспечения мульти-медиа демонстраций на основе Microsoft Office 2010 (лицензия DreamSpark Premium Electronic Software Delivery (3 years); Renewal по сублицензионному

договору №Tr000083174 от 12.04.2016);

- практические занятия по дисциплине проводятся с использованием программного обеспечения:

Netbeans IDE 7.0.1 (свободно распространяемое ПО)

Fire fox 14 (свободно распространяемое ПО)

Microsoft Office 2010 (лицензия DreamSpark Premium Electronic Software Delivery (3 years) Renewal по сублицензионному договору №Tr000083174 от 12.04.2016)

свободные видеоредакторы Kino, Kdenlive, VideoLAN Movie Creator.

свободный аудиоредактор Audacity, программы для синтеза речи Espeak, и звука FluidSynth.

программы скринкастинга RecordMyDesktop, XVidCap Screen Capture, VLC Media Player.

5.2.1 Современные профессиональные базы данных и информационные справочные системы.

Перечень СПБД и ИСС по дисциплине

1. Хаб «JavaScript» – <https://habr.com/ru/hub/javascript/>
2. MDN web docs. Ресурсы для разработчиков, от разработчиков. – <https://developer.mozilla.org/ru/>
3. Хаб «jQuery» – <https://habr.com/ru/hub/jquery/>
4. html academy – интерактивные онлайн-курсы – <https://htmlacademy.ru/>
5. Хаб «ReactJS» – <https://habr.com/ru/hub/reactjs/>
6. Хаб «Canvas» – <https://habr.com/ru/hub/canvas/>
7. The best of JavaScript, HTML and CSS. Список наиболее популярных проектов с открытым исходным кодом, связанных с веб-платформой и Node.js – <https://bestofjs.org/>

6 Иные сведения и (или) материалы.

6.1.Примерные темы письменных учебных работ

Примерные темы рефератов

1. Языки реализации клиентских сценариев.
2. JavaScript - язык разработки клиентских веб-приложений.
3. JavaScript - язык разработки серверных приложений.
4. JavaScript - язык разработки мобильных приложений.
5. JavaScript - язык разработки виджетов.
6. JavaScript - язык разработки прикладного программного обеспечения.

Примерные темы программных проектов

1. Создание интерактивного меню средствами JavaScript.
2. Организация контекстного поиска средствами JavaScript.
3. Управление окнами браузера средствами JavaScript.
4. Создание интерактивной галереи средствами JavaScript.
5. Управление устройствами ввода средствами JavaScript.
6. Динамический вывод текста средствами JavaScript.
7. Применение к тексту визуальных эффектов средствами JavaScript (управление яркостью, цветом фона страниц, реализация эффекта прозрачности).
8. Защита веб-страницы средствами JavaScript (ограничение доступа к веб-странице, запрет выделения фрагментов страницы, защита от копирования).

6.2. Примерные вопросы и задания / задачи для промежуточной аттестации

Таблица 5 - Примерные теоретические вопросы и практические задания / задачи к зачету

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
----------------	---------------------------------	---

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
1. Введение в веб-разработку		
1.1 Основы веб-проектирования	1. История создания и развития мультипарадигменного языка JavaScript. Области применения JavaScript. 2. Справочники по JavaScript (Mozilla Developer Network, MSDN, Safari Developer Library и др.). Спецификации по JavaScript (ECMAScript, HTML/DOM/CSS).	1. Спроектируйте структуру веб-приложения по теме «Учебник JavaScript» 2. Спроектируйте структуру веб-приложения по теме «Сборник примеров и задач JavaScript»
1.2 Создание документов HTML	3. Синтаксис и структура HTML. Структура документа HTML . Форматирование и выравнивание текста. Списки. Ссылки. 4. HTML Связывание страниц. Графика на Web-страницах. Таблицы. Формы .	3. Разработайте для веб-приложения «Учебник JavaScript» следующие HTML-документы: главная страница, оглавление, информационная страница. 4. Разработайте для веб-приложения «Сборник примеров и задач JavaScript» следующие HTML-документы: главная страница, оглавление, информационная страница.
1.3 Введение в технологии каскадных таблиц стилей	5. Технологии каскадных таблиц стилей. Способы применения CSS. Синтаксис таблиц стилей. Блочные и строковые элементы. Свойства блоков. 6. CSS.Отступы (margin). Набивка (padding). Граница (border). Обтекание блока текста. Управление цветом в CSS. Шрифт. Текст.Позиционирование.	3. Разработайте для веб-приложения «Учебник JavaScript» встроенные и внешние таблицы стилей. 4. Разработайте для веб-приложения «Сборник примеров и задач JavaScript» следующие встроенные и внешние таблицы стилей.
1.4 Введение в веб-программирование на языке JavaScript	7. Консоли разработчика в браузерах Google Chrome, Firefox, Internet Explorer, Opera, Safari. Отладка программного кода. 8. Тег SCRIPT языка HTML. Встроенные скрипты (JavaScript-код внутри страницы, внутри тега). Внешние скрипты и порядок их выполнения. Асинхронное выполнение скриптов.	7. Выполните отладку предложенного скрипта с использованием консоли разработчика в браузерах Google Chrome, Firefox. 8. Выполните отладку предложенного скрипта с использованием консоли разработчика в браузерах Internet Explorer, Opera, Safari.
2. Основы языка JavaScript		
2.1 Лексическая структура языка. Типы данных и значения.	9. Типы данных. Число «number», строка «string», булевый (логический) тип «boolean», специальное значение «null», специальное значение «undefined». 10. Объекты «object». JavaScript : оператор typeof.	9. Создайте переменную str и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'b', символ 'e'. 10. Создайте переменные a=10 и b=2. Выведите на экран их сумму, разность, произведение и частное (результат деления).
2.2 Переменные. Выражения и операторы.	11. Выражения. Переменные. 12. Унарные и бинарные операторы, присваивание, побитовые операторы.	11. Напишите скрипт, который считает количество секунд в часе, в сутках, в месяце.

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
		12. Создайте три переменные - час, минута, секунда. С их помощью выведите текущее время в формате 'час:минута:секунда'.
2.3 Инструкции	13. Базовые UI операции alert, prompt и confirm в JavaScript. 14. JavaScript: условные операторы. 15 JavaScript: операторы цикла.	13. Напишите скрипт, с помощью которого вычисляются корни квадратного уравнения. 14. Напишите скрипт, который будет находить факториал числа. Факториал (обозначается !) - это произведение (умножение) всех целых чисел, меньше данного, и его самого. Например, $4! = 1*2*3*4$. 15. С помощью цикла найдите сумму чисел от 1 до 100.
2.4 Объекты и массивы.	16. Создание объектов в JavaScript. Операции с объектом. Компактное представление объектов. 17. Создание строк в JavaScript. Свойство length, методы charAt, toLowerCase / toUpperCase, substring / substr / slice, trim и др. Кодировка Юникод. Посимвольное сравнение строк. 18. Массивы в JavaScript. Объявление массива. Внутреннее устройство массива. Перебор элементов.	16. Создайте переменную str и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'c', символ 'e'. 17. Создайте массив arr с элементами 2, 5, 3, 9. Умножьте первый элемент массива на второй, а третий элемент на четвертый. Результаты сложите, присвойте переменной result. Выведите на экран значение этой переменной. 18. Создайте двухмерный массив. Первые два ключа - это 'ru' и 'en'. Пусть первый ключ содержит элемент, являющийся массивом названий дней недели по-русски, а второй - по-английски. Выведите с помощью этого массива понедельник по-русски и среду по английски (пусть понедельник - это нулевой день).
2.5 JS Функции	19. JavaScript. Объявление функции. Параметры. Аргументы по умолчанию. Возврат значения. Выбор имени функции. 20. JavaScript. Рекурсивные функции, контекст выполнения.	19. Напишите функцию, которая возвращает квадрат числа. Число передается параметром. 20. Напишите рекурсивную функцию для вычисления суммы цифр числа.
2.6 JS Формы	21. Навигация и свойства элементов формы. Фокусировка: focus/blur. Изменение: change, input, cut, copy, paste. Формы: отправка, событие и метод submit. 22. Особые свойства, методы и события для работы с формами <form> и элементами ввода: <input>, <select> и	21. Разработайте форму в html-документе. Привяжите всем инпутам формы следующее событие - по потери фокуса каждый инпут выводит свое value в абзац с id="test". 22. . Разработайте форму в html-документе. Сделайте так, чтобы

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
	другими.	все инпуты формы по потере фокуса проверяли свое содержимое на правильное количество символов. Сколько символов должно быть в инпуте, указывается в атрибуте data-length. Если вбито правильное количество, то граница инпута становится зеленой, если неправильное - красной.
3. Клиентский JavaScript		
3.1 Документ и объекты страницы.	1. Окружение: DOM, BOM и JS. Дерево DOM. Работа с DOM из консоли. 2. Навигация по DOM-элементам. Поиск: getElement* и querySelector*.	1. Замена span на тег b без изменения текста. Написать скрипт, который находит в html-документе внутреннее содержимое тега span, а затем заменяет этот span на тег b, внутри которого будет стоять то, что было внутри span. 2. Дан HTML код. Написать скрипт, который изменяет содержимое абзацев на их порядковый номер в в html-коде.
3.2 Введение в браузерные события, общие свойства всех событий и приёмы работы с ними.	3. Браузерные события. Порядок обработки событий. Объект события. Всплытие и перехват. 4. Делегирование событий. Приём проектирования "поведение". Действия браузера по умолчанию. Генерация событий на элементах. 5. События мыши. События клавиатуры. Загрузка документа. Загрузка скриптов, картинок, фреймов.	3. Даны картинки. Привяжите к каждой картинке событие, чтобы по клику на картинку алертом выводился ее src. 4. Даны ссылки. Привяжите всем ссылкам событие - по наведению на ссылку в атрибут title запишется ее текст. 5. Даны абзацы с числами. По нажатию на абзац в нем должен появиться квадрат числа, которое он содержит.
3.3 Формы, элементы управления. Создание графических компонентов (виджетов).	6. Создание объектов, генерирующих меню, диалог и другие компоненты интерфейса. Семантическая верстка. 7. Графические компоненты (виджеты). Вёрстка графических компонентов. Шаблонизатор LoDash. Коллбэки и события на компонентах.	6. Дан html—документ с элементом div, границы которого обведены сплошной красной линией. Написать скрипт, который по щелчку мыши в любом месте документа изменяет положение элемента div, позиционируя его центр в указанное место. 7. Создайте анимацию следующего вида: движение графического компонента в окне слева направо
3.4 Cookies и механизм сохранения данных на стороне клиента	9. Обзор cookies. Сохранение cookie. Ограничения cookie. Чтение cookies. Альтернативы cookies. 10. Сохранение данных с иерархической структурой. Ограничения. Совместное использование сохраненных данных. Хранимые данные и безопасность.	9. При заходе на страницу спросите с помощью инпута день рождения пользователя. Когда он зайдет с следующий раз - напишите сколько месяцев, дней, часов, минут и секунд осталось до его дня рождения. И пусть по этим числам запуститься обратный отсчет (то есть они будут тикать). 10. Дана форма с инпутами.

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
		Пользователь вводит какие-то данные и закрывает страницу (не факт, что он заполнил всю форму). Сделайте так, чтобы при следующем заходе на страницу введенные им ранее данные стояли на своих местах.
3.5. Работа с протоколом HTTP.	11. Использование объекта XMLHttpRequest. Создание объекта запроса. Отправка запроса. Получение синхронного ответа. Обработка асинхронного ответа. Безопасность объекта XMLHttpRequest. 12. Основные утилиты для работы с запросами GET. Получение только заголовков. HTTP-метод POST. Ответы в форматах HTML, XML и JSON. Ограничение времени ожидания запроса.	11. Напишите скрипт, в котором при вводе текста в поле ввода на сервер посылается запрос и предлагаются варианты автозаполнения этого поля (с сервера). 12. Напишите скрипт, в котором осуществляется загрузка файла на сервер с индикацией прогресса.
4. Серверный JavaScript		
4.1 Основы работы с платформой Node.js	13. Платформа Node.js для создания серверных приложений на языке JavaScript. Архитектура: потоки или асинхронный ввод/вывод с управлением по событиям. 14. Настройка Node.js. Системные требования. Установка, инструменты разработки.	13. Вывести в консоль букву Z нарисованную звездочками. Размер 5X5 символов. 14. Напишите код, выводящий в консоль все чётные числа от 0 до 20.
4.2 Модули Node.	15. Модули Node. Идентификаторы модулей и пути. Локальные модули внутри приложения. Комплектация приложения с внешними зависимостями. 16. Работа с модулями Node.js.. Структура модулей. Объект global и глобальные переменные. Передача параметров приложению.	15. Создайте модуль Node.js «Калькулятор» с функциями сложения, вычитания и умножения. Используйте созданный модуль в другом файле Node.js. 16. Создайте модуль Node.js «Калькулятор» с функциями перевода числа между двоичной, восьмеричной и шестнадцатеричной системами. Используйте созданный модуль в другом файле Node.js.
4.3 Фреймворк Express	17. Создание сервера с помощью фреймворка Express. Конвейер обработки запроса и middleware. Отправка ответа. Статические файлы. Маршрутизация. Переадресация. Передача данных приложению. 18. Представления и движок представлений Handlebars. Частичные представления с использованием Express и Node.js	17. Создайте HTTP Web Server, который готовит ответ с HTTP заголовком и сообщением. 18. Используя Handlebars, напишите программный код для генерации сервером документа, содержащего динамический список из случайных значений.
4.4 Работа с серверами MongoDB и MondoDB и	19. Node.js и MongoDB. Подключение к серверу. Получение объекта базы данных на сервере.	19. Напишите скрипты для соединения с сервером MongoDB / MySQL, создания базы данных и

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
MySQL.	Получение объекта коллекции в базе данных. Взаимодействие с коллекцией (добавление, удаление, получение, изменение данных). 20. Подключение к MySQL. Создание подключения. Возможные ошибки при подключении. Закрытие подключения. Выполнение запросов к MySQL. Параметризация запросов.	таблицы в базе данных. 20. Напишите скрипты для добавления, удаления, получения и изменения данных из базы данных MongoDB / MySQL.
4.5 Паттерн MVC. Загрузка файлов на сервер.	21. Паттерн MVC. Модели. Представления. Контроллеры в паттерне MVC. Система маршрутизации. 22. Загрузка файлов на сервер с помощью пакета multer. Настройка параметров сохранения файла. Фильтрация файлов.	21. Реализуйте загрузку файлов на HTTP сервер. 22. Реализуйте загрузку файлов на HTTP сервер с использованием фильтрации.