Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35c9d50210def0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»
Кузбасский гуманитарно-педагогический институт
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»
Факультет информатики, математики и экономики

УТВЕРЖДАЮ

Декан ФИМЭ

_____ А.В. Фомина

«23» июня 2021 г.

# Рабочая программа дисциплины

## *Б1.В.02.02 Программирование на JavaScript*

Направление подготовки
***44.03.04 Профессиональное обучение (по отраслям)***

Направленность (профиль) подготовки
***Компьютерный дизайн***

Программа бакалавриата

Квалификация выпускника
***бакалавр***

Форма обучения
***очная, заочная***

Год набора 2021

Новокузнецк 2021

**Оглавление**

1 Цель дисциплины. ..... 3
1.1 Формируемые компетенции ..... 3
1.2 Индикаторы достижения компетенций ..... 3
1.3 Знания, умения, навыки (ЗУВ) по дисциплине ..... 4
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации. ..... 5
3. Учебно-тематический план и содержание дисциплины. ..... 5
3.1 Учебно-тематический план ..... 5
3.2. Содержание занятий по видам учебной работы ..... 7
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации. ..... 12
5 Учебно-методическое обеспечение дисциплины. ..... 14
5.1 Учебная литература ..... 14
5.2 Программное и информационное обеспечение освоения дисциплины. ... 14
5.2.1 Программное обеспечение ..... 14
5.3.2 Современные профессиональные базы данных и информационные справочные системы. ..... 15
6 Иные сведения и (или) материалы. ..... 15
6.1.Примерные темы письменных учебных работ..... 15
6.2. Примерные вопросы и задания / задачи для промежуточной аттестации 15

# 1 Цель дисциплины

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП): ПК-4.

Содержание компетенций как планируемых результатов обучения по дисциплине см. таблицы 1 и 2.

## 1.1 Формируемые компетенции

Таблица 1 - Формируемые дисциплиной компетенции

| Наименование вида компетенции | Наименование категории (группы) компетенций | Код и название компетенции |
|---|---|---|
| *профессиональная* | | ПК-4 Способен выполнять деятельность и (или) демонстрировать элементы осваиваемой обучающимися деятельности, предусмотренной программой учебного предмета, курса, дисциплины (модуля), практики |

## 1.2 Индикаторы достижения компетенций

Таблица 2 – Индикаторы достижения компетенций, формируемые дисциплиной

| Код и название компетенции | Индикаторы достижения компетенции по ОПОП | Дисциплины и практики, формирующие компетенцию ОПОП |
|---|---|---|
| ПК-2 Способен обучать профессиональной деятельности в сфере компьютерного дизайна и проектирования веб сайтов | ПК 2.1 Демонстрирует владение методами работы над дизайн-проектами объектов визуальной информации; владение композиционными приемами и стилистическими особенностями проектируемого объекта визуальной информации.<br>ПК 2.2 Демонстрирует владение методами компьютерного дизайна в целом или отдельных элементов по определенному ранее визуальному стилю, создание визуальных объектов, включая разработку их метафор.<br>ПК 2.3 Демонстрирует методы использования программных и аппаратных средств для создания объектов компьютерного дизайна. | Б1.О.13 Основы компьютерного дизайна<br>Б1.О.13.01 Графический дизайн<br>Б1.О.13.02 Веб-дизайн<br>Б1.О.13.03 Компьютерная графика в веб-дизайне<br>Б1.О.13.04 Трехмерное моделирование<br>Б1.О.13.05 Проектирование веб-сайтов<br>Б1.В.02 Инструментальные средства компьютерного дизайна<br>Б1.В.02.01 Программное обеспечение для обработки растровых изображений<br>Б1.В.02.02 Программирование на JavaScript<br>Б1.В.02.03 Видеомонтаж<br>Б1.В.02.ДВ.01.01 Веб-дизайн в рекламе<br>Б1.В.02.ДВ.01.02 Мультимедиа в веб-дизайне<br>Б1.В.02.ДВ.02.01 XML в веб-дизайне<br>Б1.В.02.ДВ.02.02 Эргономика в веб-дизайне<br>Б2.О.02(У) Проектно-технологическая<br>Б2.О.03(У) Проектная<br>Б2.О.04(П) Педагогическая<br>Б2.О.05(Пд) Преддипломная<br>Б2.В.01(У) Технологическая<br>Б3.О.01(Г) Государственный экзамен (междисциплинарный)<br>Б3.О.02(Д) Выпускная квалификационная работа |

## 1.3 Знания, умения, навыки (ЗУВ) по дисциплине

Таблица 3 – Знания, умения, навыки, формируемые дисциплиной

| Код и название компетенции | Индикаторы достижения компетенции, закрепленные за дисциплиной | Знания, умения, навыки (ЗУВ), формируемые дисциплиной |
|---|---|---|
| ПК-2 | ПК 2.3 Демонстрирует методы использования программных и аппаратных средств для создания объектов компьютерного дизайна. | Знать:<br>– терминологию, используемую в клиент-серверных и веб-приложениях;<br>– синтаксис языка JavaScript;<br>– встроенные функции языка JavaScript;<br>– модель взаимодействия скриптов и содержимого веб-страниц;<br>– принципы объектно-ориентированного дизайна и проектирования с использованием JavaScript.;<br>Уметь:<br>– формулировать требования к создаваемым программным комплексам;<br>– формировать архитектуру Web-приложений для информатизации предприятий и организаций;<br>– создавать клиентские скрипты на языке javascript;<br>– тестировать и отлаживать клиентские скрипы.<br>Владеть:<br>– инструментальными средствами написания, отладки и тестирования JavaScript-кода;<br>– фреймворками взаимодействия клиентских и серверных частей приложений;<br>– фреймворками автоматизации тестирования приложений. |

## 2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.

Таблица 4 – Объем и трудоемкость дисциплины по видам учебных занятий

| Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах | Объём часов по формам обучения | | |
|---|---|---|---|
| | ОФО | ЗФО | ОЗФО |
| 1 Общая трудоемкость дисциплины | 288 | 288 | |
| 2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего) | 124 | 58 | |
| Аудиторная работа (всего): | 124 | 58 | |
| в том числе: | | | |
| лекции | 40 | 20 | |
| практические занятия, семинары | | | |
| практикумы | | | |
| лабораторные работы | 84 | 38 | |
| 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) | 128 | 217 | |
| 4 Промежуточная аттестация обучающегося:<br>4 семестр – зачет,<br>5 семестр - экзамен | <br><br>36 | <br>4<br>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 трудоёмкость *(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 7** | | | | | | | | | |
| ***1. Введение в веб-разработку*** | | | | | | | | | |
| 1.1 | Основы веб-проектирования | **8** | 2 | 2 | 4 | 1 | 2 | 5 | ПР-6 – отчет по лабораторным работам<br>ПР-4 реферат |
| 1.2 | Создание документов HTML | **8** | 2 | 2 | 4 | 1 | 2 | 5 | ПР-6 – отчет по лабораторным |

[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 трудоёмкость *(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. | | |
| | | | | | | | | | работам |
| 1.3 | Введение в технологии каскадных таблиц стилей | **8** | 2 | 2 | 4 | 1 | 2 | 5 | ПР-6 – отчет по лабораторным работам |
| 1.4 | Введение в веб-программирование на языке JavaScript | **12** | 2 | 4 | 6 | 1 | 2 | 9 | ПР-6 – отчет по лабораторным работам |
| ***2. Основы языка JavaScript*** | | | | | | | | | |
| 2.1 | Лексическая структура языка. Типы данных и значения. | **12** | 2 | 4 | 6 | 1 | 2 | 9 | ПР-6 – отчет по лабораторным работам |
| 2.2 | Переменные. Выражения и операторы. | **12** | 2 | 4 | 6 | 1 | 2 | 9 | ПР-6 – отчет по лабораторным работам |
| 2.3 | Инструкции | **12** | 2 | 4 | 6 | 1 | 2 | 9 | ПР-6 – отчет по лабораторным работам |
| 2.4 | Объекты и массивы. | **12** | 2 | 4 | 6 | 1 | 4 | 7 | ПР-6 – отчет по лабораторным работам |
| 2.5 | JS Функции | **12** | 2 | 4 | 6 | 1 | 4 | 7 | ПР-6 – отчет по лабораторным работам |
| 2.6 | JS Формы | **12** | 2 | 4 | 6 | 1 | 4 | 7 | ПР-6 – отчет по лабораторным работам |
| | Промежуточная аттестация – ***зачет*** | | | | | | | | УО-3 – зачет |
| **ИТОГО по семестру 7** | | **108** | **20** | **34** | **54** | **10** | **26** | **72** | |
| **Семестр 8** | | | | | | | | | |
| ***3. Клиентский JavaScript*** | | | | | | | | | |
| 3.1 | Документ и объекты страницы. | **16** | 2 | 6 | 8 | 1 | 2 | 13 | ПР-6 – отчет по лабораторным работам |
| 3.2 | Введение в браузерные события, общие свойства всех событий и приёмы работы с ними. | **16** | 2 | 6 | 8 | 1 | 2 | 13 | ПР-6 – отчет по лабораторным работам |
| 3.3 | Формы, элементы управления. Создание графических компонентов (виджетов). | **16** | 2 | 6 | 8 | 1 | 2 | 13 | ПР-6 – отчет по лабораторным работам |
| 3.4 | Cookies и механизм сохранения данных на стороне клиента | **16** | 2 | 6 | 8 | 1 | 2 | 13 | |
| 3.5 | Работа с протоколом HTTP. | **16** | 2 | 6 | 8 | 1 | 2 | 13 | |
| ***4. Серверный JavaScript*** | | | | | | | | | |
| 4.1 | Основы работы с платформой Node.js | **12** | 2 | 4 | 6 | 1 | 2 | 9 | ПР-6 – отчет по лабораторным работам |
| 4.2 | Модули Node. | **12** | 2 | 4 | 6 | 1 | | 11 | ПР-6 – отчет по лабораторным работам |
| 4.3 | Фреймворк Express | **12** | 2 | 4 | 6 | 1 | | 11 | ПР-6 – отчет по лабораторным работам |

| № недели п/п | Разделы и темы дисциплины по занятиям | Общая трудоёмкость *(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. | | |
| 4.4 | Работа с серверами MondoDB и MySQL. | **14** | 2 | 4 | 8 | 1 | | 13 | ПР-6 – отчет по лабораторным работам |
| 4.5 | Паттерн MVC. Загрузка файлов на сервер. | **14** | 2 | 4 | 8 | 1 | | 13 | ПР-6 – отчет по лабораторным работам |
| | Промежуточная аттестация - *экзамен* | **36** | | | | | | **9** | УО-4 – экзамен |
| **ИТОГО по семестру 8** | | **180** | **20** | **50** | **74** | **10** | **12** | **149** | |
| | **Всего:** | **288** | **40** | **84** | **128** | **20** | **38** | **217** | |

## 3.2. Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

| № п/п | Наименование раздела, темы дисциплины | Содержание занятия |
|---|---|---|
| | **Семестр 7** | |
| | *Содержание лекционного курса* | |
| ***1. Введение в веб-разработку*** | | |
| 1.1 | Основы веб-проектирования | Internet-приложения. Основные определения и понятия. Типы Internet-приложений. Архитектурные шаблоны Web-приложений. Принципы функционирования Web-приложений. Web-приложения в сетях Intranet. Web-приложения с модулями расширения клиентской и серверной части. Архитектура Web-приложений, публикующих базы данных. Internet-приложения. Классификация Web-приложений. Двухуровневые Web-приложения. Трехуровневые Web-приложения. Многоуровневые Web-приложения. |
| 1.2 | Создание документов HTML | Синтаксис и структура HTML. Структура документа HTML. Форматирование и выравнивание текста. Списки. Ссылки. Связывание страниц. Графика на Web-страницах. Таблицы. Формы. |
| 1.3 | Введение в технологии каскадных таблиц стилей | Технологии каскадных таблиц стилей. Способы применения CSS. Синтаксис таблиц стилей. Блочные и строковые элементы. Свойства блоков. Отступы (margin). Набивка (padding). Граница (border). Обтекание блока текста. Управление цветом в CSS. Шрифт. Текст. Позиционирование. |
| 1.4 | Введение в веб-программирование на языке JavaScript | История создания и развития мультипарадигменного языка JavaScript. Области применения JavaScript: веб-приложения (AJAX, Comet, браузерные операционные системы), пользовательские скрипты в браузере, серверные приложения, мобильные приложения, виджеты, прикладное программное обеспечение. JavaScript и альтернативные браузерные технологии. Справочники и спецификации JavaScript |
| ***2. Основы языка JavaScript*** | | |
| 2.1 | Лексическая структура языка. Типы данных и значения. | Основные синтаксические конструкции языка программирования JavaScript. Типы данных (число «number», строка «string», булевый (логический) тип «boolean», специальное значение «null», специальное значение «undefined», объекты «object». |
| 2.2 | Переменные. Выражения | Основные синтаксические конструкции языка |

| № п/п | Наименование раздела, темы дисциплины | Содержание занятия |
|---|---|---|
| | и операции. | программирования JavaScript. Переменные. Выражения. Унарные и бинарные операции, присваивание. Арифметические операции. Операции сравнения. Логические операции. Побитовые операции. |
| 2.3 | Инструкции. | Базовые UI операции alert, prompt и confirm. Условные операторы. Операторы цикла: оператор do…while, while, for, for…in. Оператор continue, break, switch. |
| 2.4 | JS Функции. | Функциональный подход к программированию на языке JavaScript. Объявление функции. Локальные переменные. Внешние переменные. Параметры. Аргументы по умолчанию. Возврат значения. Выбор имени функции. Функциональные выражения. Рекурсия. Рекурсивные функции, контекст выполнения. |
| 2.5 | Объекты и массивы. | Основные свойства и методы структур данных в JavaScript. Числа. Объект Number. Объект Math. Объект Date. Строки. Объект String. Объект Object. Объекты как ассоциативные массивы. Объекты: перебор свойств, передача по ссылке. Массивы с числовыми индексами. |
| 2.6 | JS Формы. | Различные приемы работы сценариев JavaScript с HTML-формами. Свойства форм. Элементы форм. Кнопки (BUTTON, RESET, SUBMIT). Флажок (CHECKBOX). Переключатель (RADIO). Список (SELECT). Поле ввода (TEXT). Текстовая область (TEXTAREA). Поле ввода пароля (PASSWORD). Передача данных между формами на различных страницах. |
| *Содержание лабораторных занятий* | | |
| ***1. Введение в веб-разработку*** | | |
| 1.1 | Основы веб-проектирования | Проектирование структуры вебсайта. Структура статического веб-сайта. Структура динамического веб-сайта. |
| 1.2 | Создание документов HTML | Создание главной страницы вебсайта. Создание карты сайта. |
| 1.3 | Введение в технологии каскадных таблиц стилей | Создание внешних таблиц стилей для основных разделов вебсайта. |
| 1.4 | Введение в веб-программирование на языке JavaScript | Браузерные инструменты разработки . Встроенные инструменты разработки браузеров Google Chrome, Firefox, Internet Explorer, Opera, Safari. Доступ к консоли разработчика. Отладка программного кода. |
| 1.4 | Введение в веб-программирование на языке JavaScript | Встроенные и внешние скрипты. Тег SCRIPT языка HTML. Встроенные скрипты (JavaScript-код внутри страницы, внутри тега). Внешние скрипты и порядок их выполнения. Асинхронное выполнение скриптов. |
| ***2. Основы языка JavaScript*** | | |
| 2.1 | Линейные алгоритмы. | Ввод-вывод данных (alert, prompt и confirm). Основные операторы (присваивание, инкремент, декремент). Построение выражений, вычисление значений. |
| 2.1 | Ветвление. | Организация ветвлений в программах. Операторы if, ‘?’? switch. |
| 2.2 | Циклические алгоритмы | Разработка циклических алгоритмов с использованием инструкций for, for…in |
| 2.2 | Циклические алгоритмы | Разработка циклических алгоритмов с использованием инструкции while, do…while. |
| 2.3 | Функции. | Создание функций и функциональных выражений. |
| 2.3 | Рекурсивные функции. | Создание рекурсивных функций. |
| 2.4 | Алгоритмы работы со | Создание строк. Свойство length, методы charAt, toLowerCase / |

| № п/п | Наименование раздела, темы дисциплины | Содержание занятия |
|---|---|---|
| | строками. | toUpperCase, substring /substr / slice, trim и др. |
| 2.4 | Алгоритмы работы со строками. | Кодировка Юникод. Посимвольное сравнение строк. |
| 2.5 | Объекты как ассоциативные массивы. | Создание объектов. Операции с объектом. Компактное представление объектов. |
| 2.5 | Массивы с числовыми индексами. | Объявление массива. Методы pop/push, shift/unshift. Внутреннее устройство массива. Перебор элементов. Особенности работы length. Создание вызовом new Array. Внутреннее представление массивов. |
| 2.6 | JS Формы. | Создание форм. Получение значения из формы. Установка значения в форме. |
| 2.6 | JS Формы. | Проверка данных формы (проверка валидности формы на стороне клиента). Использование втсроенной проверки формы. Примеры встроенных форм валидации. Проверка форм с использованием JavaScript/ |
| **Семестр 8** | | |
| *Содержание лекционного курса* | | |
| ***3. Клиентский JavaScript*** | | |
| 3.1 | Документ и объекты страницы. | Окружение: DOM, BOM и JS. Объектная модель документа. Возможности объектной модели документа. Дерево DOM. Родительские и дочерние элементы. Доступ к элементам страницы. Работа с DOM из консоли. Навигация по DOM-элементам. Поиск: getElement* и querySelector*. Внутреннее устройство поисковых методов. Свойства узлов: тип, тег и содержимое. Современный DOM: полифиллы. Атрибуты и DOM-свойства. Методы contains и compareDocumentPosition. Добавление и удаление узлов. Мультивставка: insertAdjacentHTML и DocumentFragment. Метод document.write. Стили, getComputedStyle. Размеры и прокрутка элементов. Размеры и прокрутка страницы. Координаты в окне. Координаты в документе. |
| 3.2 | Введение в браузерные события, общие свойства всех событий и приёмы работы с ними. | Введение в браузерные события. Модель событий. Порядок обработки событий. Объект события. Всплытие и перехват. Делегирование событий. Приём проектирования "поведение". Действия браузера по умолчанию. Генерация событий на элементах. |
| 3.3 | Формы, элементы управления. Создание графических компонентов (виджетов). | Навигация и свойства элементов формы. Фокусировка: focus/blur. Изменение: change, input, cut, copy, paste. Формы: отправка, событие и метод submit.<br>Графические компоненты (виджеты). Вёрстка графических компонентов. Шаблонизатор LoDash. Коллбэки и события на компонентах |
| 3.4 | Cookies и механизм сохранения данных на стороне клиента | Обзор cookies. Сохранение cookie. Ограничения cookie. Чтение cookies. Примеры работы с cookie. Альтернативы cookies. Механизм сохранения userData в IE. Сохранение данных с иерархической структурой. Ограничения. Совместное использование сохраненных данных. Механизм сохранения SharedObject подключаемого Flash-модуля. Хранимые данные и безопасность. |
| 3.5 | Работа с протоколом HTTP. | Использование объекта XMLHttpRequest. Создание объекта запроса. Отправка запроса. Получение синхронного ответа. Обработка асинхронного ответа. Безопасность объекта XMLHttpRequest. Примеры и утилиты с объектом XMLHttpRequest. Основные утилиты для работы с запросами GET. Получение только заголовков. HTTP-метод POST. |

| № п/п | Наименование раздела, темы дисциплины | Содержание занятия |
|---|---|---|
| | | Ответы в форматах HTML, XML и JSON. Ограничение времени ожидания запроса. Ajax и динамические сценарии. Пример применения Ajax. Одностраничные приложения. Удаленное взаимодействие. Взаимодействие с протоколом HTTP с помощью тега <script>. |
| ***4. Серверный JavaScript*** | | |
| 4.1 | Основы работы с платформой Node.js | Серверный JavaScript. Платформа Node.js для создания серверных приложений на языке JavaScript. Архитектура: потоки или асинхронный ввод/вывод с управлением по события. Производительность и использование процессора. Настройка Node. Системные требования. Установка, инструменты разработки. |
| 4.2 | Модули Node. | Модули. Идентификаторы модулей и пути. Локальные модули внутри приложения. Комплектация приложения с внешними зависимостями. Системные модули в каталогах, перечисленных в массиве require.paths. Составные модули – модули-каталоги. Определение конструкторов и объектов в модуле. Работа с модулями. Структура модулей. Объект global и глобальные переменные. Передача параметров приложению. NPM. Package.json. Установка модулей. Определение команд. Асинхронность в Node.js. Работа с файлами. События. Менеджер пакетов для Node (npm). Формат npm-пакета. Поиск npm-пакетов. Команды npm. Версии и диапазоны версий пакета. Спецификация CommonJS. |
| 4.3 | Фреймворк Express | Создание сервера с помощью фреймворка Express. Конвейер обработки запроса и middleware. Отправка ответа. Статические файлы. Маршрутизация. Переадресация. Передача данных приложению. Параметры строки запроса. POST-запросы и отправка форм. Параметры маршрута. Router. JSON и AJAX. Представления и движок представлений Handlebars. Частичные представления с использованием Express и Node.js. |
| 4.4 | Работа с серверами MondoDB и MySQL. | Node.js и MongoDB. Подключение к серверу. Получение объекта базы данных на сервере. Получение объекта коллекции в базе данных. Взаимодействие с коллекцией (добавление, удаление, получение, изменение данных). Express и MongoDB. ODM-библиотека (Object Data Modelling) Mongoose для работы с MongoDB.<br>Подключение к MySQL. Создание подключения. Возможные ошибки при подключении. Закрытие подключения. Выполнение запросов к MySQL. Параметризация запросов. Promise API. Пулы подключений . MySQL и Express. Основные операции с данными. Создание базы данных. Создание таблиц. Добавление данных. Добавление множества значений. Получение данных. Фильтрация данных. Обновление. Удаление. |
| 4.5 | Паттерн MVC. Загрузка файлов на сервер. | Паттерн MVC. Модели. Представления. Контроллеры в паттерне MVC. Система маршрутизации.<br>Загрузка файлов на сервер с помощью пакета multer. Настройка параметров сохранения файла. Фильтрация файлов. |
| *Содержание лабораторных занятий* | | |
| ***3. Клиентский JavaScript*** | | |
| 3.1 | Документ и объекты страницы. | Доступ к элементу. Выбранные элементы $0 $1… Методы поиска элементов в консоли. Console API Reference для Chrome и Command Line API для Firebug. |
| 3.1 | Документ и объекты | Поиск html-элементов по id, tag name, class name, селектору |

| № п/п | Наименование раздела, темы дисциплины | Содержание занятия |
|---|---|---|
| | страницы. | css. Добавление html-элементов в документ.Изменение содержимого html-элемента, изменение значения html-атрибутов. Изменение стилей html-элементов. |
| 3.1 | Документ и объекты страницы. | Анимация с JavaScript. Создание простой веб-страницы. Создание контейнера для анимации. Создание стилей анимируемых элементов. Разработка программного кода для анимации html-элемента. |
| 3.2 | Введение в браузерные события, общие свойства всех событий и приёмы работы с ними. | События мыши. События клавиатуры. Написание функций-обработчиков событий мыши и клавиатуры. Вызов функций-обработчиков событий. Связывание событий с html-элементами с использованием JavaScript. |
| 3.2 | Введение в браузерные события, общие свойства всех событий и приёмы работы с ними. | Загрузка документа. События onload и onunload. Загрузка скриптов, картинок, фреймов. |
| 3.2 | Введение в браузерные события, общие свойства всех событий и приёмы работы с ними. | JavaScript HTML DOM EventListener. Метод addEventListener. Добавление обработчика событий к элементу. Добавление нескольких обработчиков событий к элементу. Добавление обработчика событий к объекту window. Передача параметров. Метод removeEventListener. |
| 3.3 | Формы, элементы управления. Создание графических компонентов (виджетов). | Особые свойства, методы и события для работы с формами <form> и элементами ввода: <input>, <select> и другими. |
| 3.3 | Формы, элементы управления. Создание графических компонентов (виджетов). | Создание виджетов. Создание объектов, генерирующих меню, диалог и другие компоненты интерфейса. Семантическая верстка. |
| 3.3 | Формы, элементы управления. Создание графических компонентов (виджетов). | Создание окон и фреймов. Открытие окон и методы window. Общение между окнами и фреймами. Кросс-доменные ограничения и их обход. Общение окон с разных доменов: postMessage. Привлечение внимания к окну. Атака Clickjacking и защита от неё. |
| 3.4 | Cookies и механизм сохранения данных на стороне клиента | Хранение данных в браузере. document.cookie. Чтение из document.cookie. Запись в document.cookie. Настройки куки |
| 3.4 | Cookies и механизм сохранения данных на стороне клиента | Функции для работы с куки: getCookie(name), setCookie(name, value, options), deleteCookie(name). |
| 3.4 | Cookies и механизм сохранения данных на стороне клиента | Объекты веб-хранилища localStorage и sessionStorage. База данных IndexedDB. |
| 3.5 | Работа с протоколом HTTP. | Метод fetch. Заголовки ответа. Заголовки запроса. POST-запросы. Отправка изображения. Fetch: ход загрузки. Fetch: прерывание запроса. |
| 3.5 | Работа с протоколом HTTP. | Отправка HTML-форм. Отправка простой формы. Методы объекта FormData. Отправка формы с файлом. Отправка формы с Blob-данными. |
| 3.5 | Работа с протоколом HTTP. | Fetch: запросы на другие сайты. Простые запросы. CORS для простых запросов. Заголовки ответа. «Непростые» запросы. Авторизационные данные. |
| ***4. Серверный JavaScript*** | | |
| 4.1 | Основы работы с платформой Node.js | Разработка простейшего приложения для NodeJS. |

| № п/п | Наименование раздела, темы дисциплины | Содержание занятия |
|---|---|---|
| 4.1 | Основы работы с платформой Node.js | Модуль http для работы с сервером и протоколом http в Node.js. Создание сервера. Параметр request. Параметр response. Маршрутизация. Переадресация. Отправка статических файлов. Шаблоны. |
| 4.2 | Модули Node. | Использование встроенных модулей Node.js. |
| 4.2 | Модули Node. | Создание модуля Node.js. Тестирование модуля. Сохранение локального модуля как зависимости. Привязка локального модуля. |
| 4.3 | Фреймворк Express | Создание простейшего приложения с использованием фреймворка Express. Создание сервера с помощью фреймворка Express. Конвейер обработки запроса и middleware. Отправка ответа. Статические файлы. Маршрутизация. Переадресация. |
| 4.3 | Фреймворк Express | Создание простейшего приложения с использованием фреймворка Express. Передача данных приложению. Параметры строки запроса. POST-запросы и отправка форм. Параметры маршрута. Router. |
| 4.4 | Работа с сервером MondoDB. | Создание простейшего приложения, реализующего основные операции с данными. Создание базы данных. Создание таблиц. Добавление данных. Добавление множества значений. Получение данных. Фильтрация данных. Обновление. Удаление. |
| 4.4 | Работа с сервером MySQL. | Создание простейшего приложения, реализующего основные операции с данными. Создание базы данных. Создание таблиц. Добавление данных. Добавление множества значений. Получение данных. Фильтрация данных. Обновление. Удаление. |
| 4.5 | Паттерн MVC. | Разработка простейшего приложения с использованием паттерна MVC. |
| 4.5 | Загрузка файлов на сервер. | Загрузка файлов на сервер с помощью пакета multer. Настройка параметров сохранения файла. Фильтрация файлов |

## 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 7.

Таблица 7 - Балльно-рейтинговая оценка результатов учебной работы

**7 Семестр**

| Составляющие | Сумма баллов | Учебная деятельность студента | Оценка в аттестации | Баллы (17 недель) |
|---|---|---|---|---|
| Текущая учебная деятельность | **80** | Посещение лекций (7 занятий). | **1 балл** (присутствие на лекции)<br>**2 балла** (активная работа, конспектирование) | 7 - 14 |
| | | Лабораторные работы (28 работ). | **1 балл** (посещение занятия, выполнение работы на 51-65%)<br>**2 балла** (существенный вклад на занятии относительно всей группы, самостоятельность при выполнении работы, выполнение работы на 85,1- | 28 - 56 |

| | | | | |
|---|---|---|---|---|
| | | | 100%) | |
| | | Реферат | **6 баллов** (пороговое значение)<br>**10 баллов** (максимальное значение) | 6 - 10 |
| | | Разработка проекта | **10 баллов** (пороговое значение)<br>**20 баллов** (максимальное значение) | 10 - 20 |
| **Итого по текущей работе в семестре** | | | | 51 - 100 |
| Промежуточ-ная аттестация (зачет) | **20** | Теоретический вопрос 1. | **1 балл** (пороговое значение)<br>**5 баллов** (максимальное значение) | 1 - 5 |
| | | Теоретический вопрос 2. | **1 балл** (пороговое значение)<br>**5 баллов** (максимальное значение) | 1 - 5 |
| | | Защита проекта | **3 балла** (пороговое значение)<br>**10 баллов** (максимальное значение) | 3 - 10 |
| **Итого по промежуточной аттестации (зачету)** | | | | 5 – 20 б. |
| **Суммарная оценка по дисциплине:** | | Сумма баллов текущей и промежуточной аттестации | | 51 – 100 б. |

**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)<br>**2 балла** (активная работа, конспектирование) | 12 - 24 |
| | | Лабораторные работы (18 работ). | **1 балл** (посещение занятия, выполнение работы на 51-65%)<br>**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)<br>**10 баллов** (максимальное значение) | 6 - 10 |
| | | Разработка проекта | **15 баллов** (пороговое значение)<br>**30 баллов** (максимальное значение) | 15-30 |
| **Итого по текущей работе в семестре** | | | | 51 - 100 |
| Промежуточ-ная аттестация (экзамен) | **40** | Теоретический вопрос 1. | **5 баллов** (пороговое значение)<br>**10 баллов** (максимальное значение) | 5 - 10 |
| | | Теоретический вопрос 2. | **5 баллов** (пороговое значение)<br>**5 баллов** (максимальное значение) | 5 - 10 |
| | | Защита проекта | **10 баллов** (пороговое значение)<br>**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. Слинкин. — 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 Программное и информационное обеспечение освоения дисциплины.

### 5.2.1 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 Screeen Capture, VLC Media Player.

### 5.2.2 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

***Семестр 7***

Таблица 9 - Примерные теоретические вопросы и практические задания / задачи к зачету

| 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.<br>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»<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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».<br>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'.<br>10. Создайте переменные a=10 и b=2. Выведите на экран их |

| Разделы и темы | Примерные теоретические вопросы | Примерные практические задания / задачи |
|---|---|---|
| | | сумму, разность, произведение и частное (результат деления). |
| 2.2 Переменные. Выражения и операторы. | 11. Выражения. Переменные.<br>12. Унарные и бинарные операторы, присваивание, побитовые операторы. | 11. Напишите скрипт, который считает количество секунд в часе, в сутках, в месяце.<br>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.<br>14. JavaScript: условные операторы.<br>15 JavaScript: операторы цикла. | 13. Напишите скрипт, с помощью которого вычисляются корни квадратного уравнения.<br>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.<br>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.<br>17. Создание строк в JavaScript. Свойство length, методы charAt, toLowerCase / toUpperCase, substring /substr / slice, trim и др. Кодировка Юникод. Посимвольное сравнение строк.<br>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'.<br>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.<br>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.<br>20. JavaScript. Рекурсивные функции, контекст выполнения. | 19. Напишите функцию, которая возвращает квадрат числа. Число передается параметром.<br>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, | 21. Разработайте форму в hml-документе. Привяжите всем инпутам формы следующее |

| Разделы и темы | Примерные теоретические вопросы | Примерные практические задания / задачи |
|---|---|---|
| | paste. Формы: отправка, событие и метод submit.<br>22. Особые свойства, методы и события для работы с формами <form> и элементами ввода: <input>, <select> и другими. | событие - по потери фокуса каждый инпут выводит свое value в абзац с id="test".<br>22. . Разработайте форму в hml-документе. Сделайте так, чтобы все инпуты формы по потери 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. |

***Семестр* 8**

Таблица 9 - Примерные теоретические вопросы и практические задания / задачи к экзамену

| Разделы и темы | Примерные теоретические вопросы | Примерные практические задания / задачи |
|---|---|---|
| 3. Клиентский JavaScript | | |
| 3.1 Документ и объекты страницы. | 1. Окружение: DOM, BOM и JS. Дерево DOM. Работа с DOM из консоли.<br>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.<br>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.<br>4. Делегирование событий. Приём проектирования "поведение". Действия браузера по умолчанию. Генерация событий на элементах.<br>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.<br>4. Даны ссылки. Привяжите всем ссылкам событие - по наведению на ссылку в атрибут title запишется ее текст.<br>5. Даны абзацы с числами. По нажатию на абзац в нем должен появится 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.<br>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.<br>7. Создайте анимацию следующего вида: движение графического |

| Разделы и темы | Примерные теоретические вопросы | Примерные практические задания / задачи |
|---|---|---|
| | | 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.<br>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).<br>10. Дана форма с инпутами. 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.<br>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).<br>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 события.<br>14. Настройка Node.js. Системные требования. Установка, инструменты разработки. | 13. Вывести в консоль букву Z нарисованную звёздочками. Размер 5X5 символов.<br><br>14. Напишите код, выодящий 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.<br>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 омдуль в другом файле Node.js.<br>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. | 17. Создайте HTTP Web Server, который готовит ответ с HTTP заголовком и сообщением. |

| Разделы и темы | Примерные теоретические вопросы | Примерные практические задания / задачи |
|---|---|---|
| | Отправка ответа. Статические файлы. Маршрутизация. Переадресация. Передача данных приложению.<br>18. Представления и движок представлений Handlebars. Частичные представления с использованием Express и Node.js | 18. Используя Handlebars, напишите программный код для генерации сервером документа, содержащего динамический список из случайных значений. |
| 4.4 Работа с серверами MondoDB и MySQL. | 19. Node.js и MongoDB. Подключение к серверу. Получение объекта базы данных на сервере. Получение объекта коллекции в базе данных. Взаимодействие с коллекцией (добавление, удаление, получение, изменение данных).<br>20. Подключение к MySQL. Создание подключения. Возможные ошибки при подключении. Закрытие подключения. Выполнение запросов к MySQL. Параметризация запросов. | 19. Напишите скрипты для соединения с сервером MongoDB / MysSQL, создания базы данных и таблицы в базе данных.<br>20. Напишите скрипты для добавления, удаления, получения и изменения данных из базы данных MongoDB / MysSQL. |
| 4.5 Паттерн MVC. Загрузка файлов на сервер. | 21. Паттерн MVC. Модели. Представления. Контроллеры в паттерне MVC. Система маршрутизации.<br>22. Загрузка файлов на сервер с помощью пакета multer. Настройка параметров сохранения файла. Фильтрация файлов. | 21. Реализуйте загрузку файлов на HTTP сервер.<br>22. Реализуйте загрузку файлов на HTTP сервер с использованием фильтрации. |

Составитель (и): Бойченко Г.Н, доцент кафедры ИОТД