

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Сахалинский государственный университет»

Кафедра информатики


УТВЕРЖДЕН

на заседании кафедры

«19 » марта 2024 г., протокол № 8

Исполняющий обязанности

заведующего кафедрой



Осипов Г.С.

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Б1.В.06 Web-технологии, языки и средства создания web-приложений

Направление подготовки
10.03.01 Информационная безопасность

профиль

*Безопасность автоматизированных систем (по отрасли или в сфере профессиональной
деятельности)*

Уровень высшего образования

БАКАЛАВРИАТ

Южно-Сахалинск
2024 г.

1. Формируемые компетенции и индикаторы их достижения по дисциплине (модулю)

Код компетенции	Содержание компетенции	Код и наименование индикатора достижения компетенции
ПКС-2	Способен решать задачи профессиональной деятельности с учетом текущего состояния и тенденций развития информационных технологий, средств технической защиты информации, сетей и систем передачи информации	<p>ПКС-2.1 Знает методы решения задач профессиональной деятельности с учетом текущего состояния и тенденций развития информационных технологий, средств технической защиты информации, сетей и систем передачи информации</p> <p>ПКС-2.2 Умеет решать задачи профессиональной деятельности с учетом текущего состояния и тенденций развития информационных технологий, средств технической защиты информации, сетей и систем передачи информации</p> <p>ПКС-2.3 Решает различные задачи профессиональной деятельности с учетом текущего состояния и тенденций развития информационных технологий, средств технической защиты информации, сетей и систем передачи информации</p>

2. Паспорт фонда оценочных средств по дисциплине (модулю)

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Тема 1. Принципы разработки Web-документов	ПКС-2	Вопросы для обсуждения по теме лекции, задания к лабораторным работам, домашнее задание
2.	Тема 2. Язык гипертекстовой разметки страниц HTML	ПКС-2	Вопросы для обсуждения по теме лекции, задания к лабораторным работам, домашнее задание, тест
3.	Тема 3. Использование стиля при оформлении сайта. Спецификации CSS	ПКС-2	Вопросы для обсуждения по теме лекции, задания к лабораторным работам домашнее задание, тест
4.	Коллоквиум	ПКС-2	Вопросы к коллоквиуму
5.	Зачет	ПКС-2	Вопросы к зачету
6.	Тема 4. Язык создания динамических страниц JavaScript	ПКС-2	Вопросы для обсуждения по теме лекции, задания к лабораторным работам, домашнее задание, тест
7.	Тема 5. Программирование на стороне сервера с использованием языка PHP	ПКС-2	Вопросы для обсуждения по теме лекции, задания к лабораторным работам, домашнее задание, тест
8.	Тема 6. Базы данных MySQL	ПКС-2	Вопросы для обсуждения по теме лекции, задания к лабораторным работам, домашнее задание, тест
9.	Коллоквиум	ПКС-2	Вопросы к коллоквиуму
10.	Экзамен	ПКС-2	Вопросы к экзамену

3. Оценочные средства

3 семестр

Лекция 1 (2 ч.). Тема: Принципы разработки Web-документов.

Вопросы для обсуждения:

1. Организация Web-сайта (Web-документов).
2. Современные технологии разработки Web-документов.
3. Язык HTML как средство создания Web-страниц.
4. Структура HTML-документа.
5. Основные элементы языка HTML.
6. Дизайн в Web-сайтах.

Лекция 2 (6 ч.). Тема: Язык гипертекстовой разметки страниц HTML.

Вопросы для обсуждения:

1. Язык гипертекстовой разметки страниц HTML: общая структура документа, абзацы, цвета, ссылки, списки, графика (графические форматы, графический объект как ссылка), таблицы.
2. Фреймы.
3. Общие подходы к дизайну сайта.
4. Разработка макета страницы.
5. Формы.

Лекция 3 (10 ч.). Тема: Использование стиля при оформлении сайта. Спецификации CSS.

Вопросы для обсуждения:

1. Использование стиля при оформлении сайта.
2. Возможности CSS.
3. Свойства текста.
4. Свойства цвета и фона.
5. Свойства шрифта.
6. Свойства блоков.
7. Свойства списков.
8. Классы. Псевдоклассы.

Лабораторное занятие 1 (2 ч.). Тема: Принципы разработки Web-документов. Базовый синтаксис HTML.

Вопросы для обсуждения:

1. Структура web-страницы. Создание простой web-страницы. Форматирование текста.
2. Использование спецсимволов. Создание списков. Вставка ссылок в документ.
3. Создание линеек. Размещение графики на web-странице: изображения, карты изображений.

Примерные задания:

1. Найти в компьютерной системе произвольный файл с расширением htm. Открыть его браузером. Одновременно открыть его текстовым редактором «Блокнот». Исследовать связь html-текста с его отображением браузером. Создать копию этого файла с другим именем. Изменить html-текст и посмотреть изменения в его отображении. Изменить текст и текстовые атрибуты отображения этого текста, изменяя html-код. Способ изменения найти самостоятельно методом проб и ошибок.
2. Создать html-файл. Добавить в него свою фотографию. Нарисовать с использования произвольного графического редактора схему места обучения (рекламный план положения СахГУ с указанием улиц, адреса, способа приезда на общественном транспорте, входа). Вставить схему в html-файл.

Лабораторное занятие 2 (6 ч.). Тема: Язык гипертекстовой разметки страниц HTML. Расширенный синтаксис HTML.

Вопросы для обсуждения:

1. Работа с таблицами.
2. Создание форм. Размещение элементов на форме.
3. Создание простого web-сайта на основе фреймов.

Примерные задания:

1. Спроектировать структуру веб-сайта по любой теме.
2. Разработать эскиз оформления веб-сайта (использовать любой графический редактор).

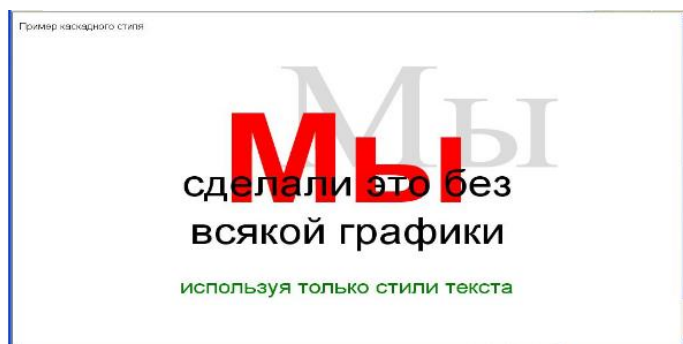
Лабораторное занятие 3 (2 ч.). Тема: Использование стиля при оформлении сайта. Спецификации CSS. Базовый синтаксис CSS3.

Вопросы для обсуждения:

1. Создание стилевых классов.
2. Форматирование блоков и списков.

Примерные задания:

1. Создайте несколько фрагментов текста и, применив к ним классы CSS, получите, например, такой результат:



2. Оформите 3 таблицы с помощью стилей CSS:

Оформление таблицы с помощью стилей CSS		
1	2	3
4	5	6

Оформление таблицы с помощью стилей CSS		
1	2	3
4	5	6

CSS таблица

Оформление таблицы с помощью стилей CSS			
	книги	журналы	газеты
Цена	20	15	3
Количество	100	100	1000
Всего	2000	1500	3000

Лабораторное занятие 4 (4 ч.). Тема: Использование стиля при оформлении сайта. Спецификации CSS. Блочные и строчные элементы.

Вопросы для обсуждения:

1. Оптимизация таблиц стилей и оформление текста.
2. Использование блочной верстки сайта.
3. Чем отличаются действия свойств display:none и visibility:hidden?
4. На веб-странице размещено изображение шириной 200 px. Как задать для него обтекание текстом по правой стороне?
5. Как поместить элемент веб-страницы (например, <p>) за видимую область экрана?

Примерные задания:

1. Выполнить верстку макета страницы с блочной структурой по разработанному эскизу для вашего сайта (см. задание к лабораторной работе № 2).
2. Создать внешние таблицы стилей (раздельные для устройств screen, print и handheld) для вашего сайта (см. задание к лабораторной работе № 2).
3. Подключить созданные таблицы к макету страницы.
4. Проверить правильность отображения веб-страниц в различных браузерах.

Лабораторное занятие 5 (4 ч.). Тема: Использование стиля при оформлении сайта. Спецификации CSS. Верстка сайта.

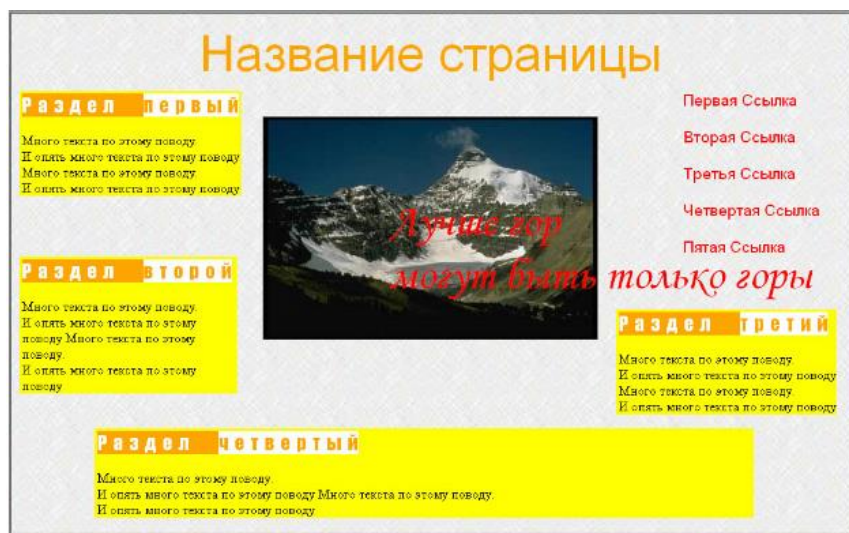
Вопросы для обсуждения:

1. Схемы позиционирования и слои.
2. Верстка сайта с использованием CSS.

Примерные задания:

1. Создайте собственную раскладку страницы с помощью блоков. Страница должна быть масштабируемой. В качестве образца можете использовать один из приведенных вариантов (см. рис.).





5 семестр

Лекция 4 (14 ч.). Тема: Язык создания динамических страниц JavaScript.

Вопросы для обсуждения:

1. Преимущества и ограничения программ, работающих на стороне клиента.
2. Понятие динамических страниц.
3. Язык JavaScript: основы синтаксиса.
4. Объектная модель HTML страницы.
5. Событийная модель DHTML: связывание событий с кодом, всплытие событий, объект Event.
6. Применение DHTML.

Лекция 5 (14 ч.). Тема: Программирование на стороне сервера с использованием языка PHP.

Вопросы для обсуждения:

1. Введение в программирование на стороне сервера на примере PHP. Принцип работы.
2. Синтаксис языка программирования PHP.
3. Переменные. Константы.
4. Операторы в PHP.
5. Циклы.
6. Массивы.
7. Работа со строками.
8. Функции в PHP. Встроенные функции.
9. Работа с датой и временем в PHP.
10. Связь PHP и HTML.
11. Создание HTML-страниц средствами PHP.

Лекция 6 (8 ч.). Тема: Базы данных MySQL.

Вопросы для обсуждения:

1. Варианты хранения информации в сети Internet.
2. Принципы хранения информации в базах данных MySQL.
3. Архитектура базы данных MySQL (таблицы, связи, триггеры).
4. Проектирование баз данных.
5. Нормализация таблиц.
6. Синтаксис запросов к базе данных.
7. Механизм работы с базами данных.
8. Управление форматами даты и времени.
9. Подключение к базе данных из PHP файла.
10. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу.

11. Передача параметров в запрос.

Лабораторное занятие 6 (6 ч.). Тема: Язык создания динамических страниц JavaScript. Основы языка JavaScript.

Вопросы для обсуждения:

1. Синтаксис. Типы данных и значения. Переменные.
2. Выражения и операторы. Инструкции-выражения.
3. Объекты и массивы.

Примерные задания:

1. Написать скрипт «Tip of the Day» (Совет дня). Скрипт должен выводить случайную строку («совет») из заданного массива строк. Скрипт разместить во внешнем файле, подключить его на все страницы вашего сайта (см. задание к лабораторной работе № 2).
2. Создать простой HTML-документ и выполнить для этого документа следующие задания:
 - Добавить два абзаца с произвольным текстом.
 - Организовать между двумя абзацами вывод приветственного сообщения в диалоговом окне, задав необходимые команды внутри тэга `<script>`.
 - Добавить команду вывода аналогичного приветственного сообщения в окно браузера после закрытия диалогового окна.
 - Сохранить документ с именем Ex1.html в рабочей папке.
3. Создать простой HTML-документ и выполнить для этого документа следующие задания:
 - Добавить два абзаца с произвольным текстом.
 - Организовать между двумя абзацами вывод приветственного сообщения в диалоговом окне, задав необходимые команды JavaScript во внешнем файле. Для этого:
 - создать новый текстовый файл,
 - поместить в него код JavaScript,
 - сохранить файл с именем main.js следующим образом: указать тип файла «Все файлы», кодировку «UTF-8».
 - Добавить ссылку на внешний скриптовый файл из рабочего HTML-документа.
 - Сохранить документ с именем Ex2.html в рабочей папке.
4. Создать простой HTML-документ и выполнить для этого документа следующие задания:
 - Сохранить документ с именем Ex3.html в рабочей папке.
 - Добавить в документ код JavaScript так, чтобы в диалоговом окне появлялось поле с надписью «Введите сюда своё имя» и со значением по умолчанию в поле «Введите имя». Для этого используйте метод `prompt(...)` объекта `window`. Для хранения введенного значения заведите новую переменную.
 - Организовать вывод введенного значения имени в окно браузера в виде: «Ваше имя <.....>».
 - Дополнить код, чтобы в новом диалоговом окне появилась надпись «Начать заново?». При положительном ответе появлялось диалоговое окно: «Не надоело?», при отказе – «Ну, и правильно!». Использовать для написания методы `alert(...)` и `confirm(...)` объекта `window`.

Лабораторное занятие 7 (6 ч.). Тема: Язык создания динамических страниц JavaScript. Модель DOM.

Вопросы для обсуждения:

1. Функции и события. Обработка событий.
2. Классы, конструкторы и прототипы.
3. Разработка сценариев JavaScript.
4. Какую директиву конфигурации Apache нужно изменить, чтобы указать в качестве стартовой страницы сайта файл default.html?
5. Какую ошибку вернет веб-сервер при попытке просмотреть содержимое каталога, для которого задана директива Options-Indexes?

6. Как изменить номер порта, на котором работает веб-сервер?
7. Что вернет веб-сервер при обращении клиента к скрипту .php, если поддержка php не включена на сервере?

Примерные задания:

1. Написать скрипт, проверяющий код защиты от автоматического постинга и вырезающий ссылки из формы ввода комментария (на странице отзывов и комментариев).
Пояснение: В ходе выполнения задания требуется написать клиентскую программу на JavaScript, которая генерирует арифметический пример, ответ на который должен дать пользователь. Другой вариант — генерация произвольной строки, которую должен воспроизвести пользователь. После того, как пользователь ввел ответ программа должна проверить его правильность.
2. Установить веб-сервер Apache.
3. Создать виртуальный хост <ваш-номер-зачетной-книжки>.local.
4. Разместить на созданном сайте все веб-документы (гипертекстовые страницы, файлы стилей, клиентские скрипты на JavaScript), созданные в предыдущих лабораторных работах.

Лабораторное занятие 8 (6 ч.). Тема: Программирование на стороне сервера с использованием языка PHP. Разработка серверных скриптов.

Вопросы для обсуждения:

1. Введение в программирование на стороне сервера на примере PHP.
2. Синтаксис языка программирования PHP. Переменные. Константы. Операторы в PHP.
3. Язык PHP. Циклы. Массивы.

Примерные задания:

1. Написать скрипт, учитывающий количество кликов по ссылкам на скачивание файлов, записывающий эти данные в файл, а затем выполняющий редирект на скачиваемый файл. Формат записи может быть, например таким: имя_файла;дата/время;ip-адрес клиента;
2. Средствами PHP создайте генератор web-страниц. Подготовьте форму, в которую бы вводился заголовок страницы, цвет фона, цвет шрифта и текст страницы. Создайте HTML-страницу с помощью этой формы.
3. Средствами PHP напишите классическую игру «Угадай число». Пусть компьютер создает случайное число и дает пользователю возможность его угадать. Сообщайте пользователю, является ли его ответ слишком большим, слишком маленьким или правильным. После того, как пользователь угадает значение, сообщите ему, сколько попыток для этого потребовалось.
4. Создайте средствами PHP простую гостевую книгу. Позвольте пользователю вводить информацию в форму, и когда он нажимает кнопку Submit, добавьте его комментарий в конец страницы.

Лабораторное занятие 9 (6 ч.). Тема: Программирование на стороне сервера с использованием языка PHP. Функциональное программирование на стороне сервера.

Вопросы для обсуждения:

1. Язык PHP. Функции в PHP.
2. Язык PHP. Связь PHP и HTML. Создание HTML-страниц средствами PHP.
3. Какой метод отправки данных из веб-формы используется по умолчанию?
4. В чем отличия при передаче данных методами GET и POST?
5. Как можно передать данные на сервер без использования форм?
6. В чем отличия загрузки файлов от передачи других данных из веб-формы?
7. Что сделает браузер при отправке почты из веб-формы (задана схема "mailto")?

Примерные задания:

1. Написать скрипт, читающий файл статистики скачиваний (см. ЛР 8) и подсчитывающий количество скачиваний по каждому файлу. Полученную информацию внедрить в страницы вашего сайта возле ссылок на соответствующие файлы (программа, руководство пользователя и прочие файлы в форматах pdf, doc, zip и т.п.).

2. Создать форму регистрации пользователя, позволяющую передать на сервер (метод GET) следующие данные:
 - Имя пользователя (обязат.)
 - Логин (обязат.)
 - Пароль и подтверждение пароля (обязат.)
 - Адрес электронной почты (обязат.)
 - Пол (не обязат.)
 - Дата рождения (не обязат.)
 - Страна и город (не обязат.)
3. Написать JavaScript, выполняющий проверку правильности заполнения полей регистрации перед отправкой на сервер.
4. Написать серверный php-скрипт, принимающий регистрационные данные и отображающий их на веб-странице.
5. Изменить метод отправки данных из формы на POST, внести соответствующие изменения в скрипты и сравнить результаты выполнения.
6. Создать форму загрузки от 1-го до 5-ти файлов на сервер и написать php-скрипт, принимающий эти файлы. Скрипт должен выводить сообщение об успешной загрузке или об ошибке в случае неудачи.

Лабораторное занятие 10 (6 ч.). Тема: Базы данных MySQL. Проектирование баз данных.

Вопросы для обсуждения:

1. Проектирование баз данных.
2. Нормализация таблиц.
3. Синтаксис запросов к базе данных.

Примерные задания:

1. Создать базу данных mysite (использовать СУБД MySQL), содержащую следующие таблицы:
 - myusers — для хранения данных о пользователях сайта (структура таблицы соответствуют данным из регистрационной формы);
 - mycontent — для хранения текстовой информации (контента), отображаемой на страницах сайта. Примерная структура таблицы: pageid, userid, lastmod, title, description, content.
2. Написать скрипт register.php, получающий данные о пользователе со страницы регистрации и записывающий эти данные в таблицу myusers.
3. Создать форму авторизации (логин и пароль) и написать скрипт login.php, сравнивающий эти данные с записями из таблицы myusers. Если пара "логин + пароль" найдена в БД, то пользователь считается авторизованным, в ином случае выполняется перенаправление на страницу регистрации.
4. Создать форму добавления контента на сайт и скрипт addpage.php, сохраняющий полученную информацию в таблице mycontent.

Примечание: Поскольку использование сессий будет рассмотрено позже, то значение userid в этом задании следует задать явно (например, в переменной).
5. Написать скрипт page.php, выбирающий данные из таблицы mycontent (по pageid) и отображающий их в виде веб-страницы.

Лабораторное занятие 11 (6 ч.). Тема: Базы данных MySQL. Взаимодействие сервера с базами данных.

Вопросы для обсуждения:

1. Подключение к базе данных из PHP файла.
2. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу.
3. Передача параметров в запрос.

Примерные задания:

1. Изменить скрипт авторизации login.php (см. ЛР 10) таким образом, чтобы для авторизованного

- пользователя открывалась новая сессия, в которой сохраняются логин и ID пользователя.
2. Переписать скрипт добавления материалов на сайт с учетом сохранения значения userid (полученного из переменной сессии) в таблице mycontent.
 3. Написать скрипт logout.php, выполняющий закрытие сессии для авторизованного пользователя.

Примерные варианты домашнего задания:

3 семестр

Домашняя работа 1. Тема: Принципы разработки Web-документов.

Задания:

1. Откройте Блокнот и наберите HTML-код для текста, состоящего из 2-х заголовков и 3-х абзацев.
2. Сохраните файл с именем 1.htm (1.html).
3. Запустите из Проводника сохраненный файл. Автоматически запустится браузер Internet Explorer.
4. Проверьте результаты, при несовпадении исправьте HTML-код страницы, сохраните и вновь проверьте результат.
5. Измените документ, разместив абзацы по центру, по левому краю и правому краю экрана.
6. Создайте документ с нумерованным и ненумерованным списками и сохраните результаты в файле 2.htm.
7. Создайте вложенный список в том же документе. Свяжите эти документы между собой, поместив на первой странице гиперссылку с именем Списки, а на второй странице гиперссылку с именем Назад.
8. Создайте документ с таблицами, сохраните его в файле 3.htm.
9. Создайте вложенную таблицу, в ячейках которой будут находиться списки, картинки и текст.

Вопросы для самоконтроля:

1. Какие теги служат для оформления текста?
2. Какие бывают списки?
3. Алгоритм оформления таблиц.
4. Назначение атрибутов тегов.
5. Как управлять браузером?

Домашняя работа 2. Тема: Язык гипертекстовой разметки страниц HTML.

Задания:

1. Выберите подходящее изображение для карты и поместите ее в текущий каталог.
2. Напишите HTML-код, создающий ссылки на предыдущие 3 файла (1.htm, 2.htm, 3.htm), используя MAP-карту. Для этого разделите выбранное изображение на 3 активные области и опишите в теге <AREA>.
3. Проверьте результат.
4. Подготовьте документ с большим количеством разделов. Создайте страницу HTML с двумя фреймами: в левом выведите список оглавления, а в правом – текст документа. Элементы оглавления должны содержать внутреннюю гиперссылку, вызывающую прокрутку документа к нужному разделу.
5. Измените HTML-код таким образом, чтобы на экране появился третий – верхний фрейм – в котором будет выводиться название документа. В заголовке документа используйте элемент H1.
6. Разработайте форму для обратной связи с посетителями сайта и напишите HTML-код со следующими элементами: текстовые поля для ввода фамилии, имени; текстовое поле для ввода пароля; текстовое поле для ввода адреса электронной почты, текстовый блок для ввода сообщения; кнопки-флажки для ввода оценки сайта или мнения о сайте.
7. Добавьте в форму кнопки очистки и отправки. Проверьте результат.

Вопросы для самоконтроля:

1. Насколько документов можно ссылаться с помощью MAP-карты?
2. Каково назначение фреймов.
3. Каково назначение формы.
4. Какой элемент формы служит для ввода пароля?
5. Какой элемент формы служит для создания меню?

Домашняя работа 3. Тема: Использование стиля при оформлении сайта. Спецификации CSS.

Задания:

1. Создайте новую страницу или используйте предыдущие страницы.
2. Используя атрибут Style, задайте встроенные стили внутри элементов <Hn>, <P>, <PRE>, , <BODY>, <A>, <HR>, <DIV>, <I>. Заметьте, каким образом осуществляется наследование стилей.
3. Используя тег <STYLE type = "text/css">, напишите код для внутренней таблицы стилей, т.е. внутри HTML-документа. Отметьте размер HTML-файла.
4. Для всех предыдущих страниц задайте одну внешнюю таблицу стилей. Тем самым можно придать всему сайту единое оформление и вид. Файл с таблицей стилей сохраните в отдельном файле с расширением .css и подключите ко всем страницам.
5. В таблице стилей задайте одинаковые правила одновременно для нескольких элементов.
6. В таблице стилей задайте одинаковые правила для родительских и дочерних элементов.
7. В таблице стилей задайте одинаковые правила для произвольных элементов HTML-документа, используя идентификаторы и классы. В CSS-файле идентификаторы обозначаются символом #, а классы начинаются с символа точки(.). В самом HTML-документе свяжите элементы с классом или идентификатором, для этого используйте в тегах атрибуты соответственно ID и CLASS.
8. Внешнюю таблицу стилей подключайте к страницам с помощью конструкции @import, а затем с помощью тега <LINK>. Отметьте размер HTML-файлов страниц.

Вопросы для самоконтроля:

1. Достоинства каскадных таблиц стилей.
2. Какие свойства можно определять в каскадных таблицах стилей?
3. Для чего необходим атрибут style?
4. Как подключаются к странице внешние таблицы?
5. Где в Html-документе располагается внутренняя таблица стилей?

Домашняя работа 4. Тема: Использование стиля при оформлении сайта. Спецификации CSS.

Задания:

1. Создайте документ с текстом и изображением.
2. Экспериментируйте с фильтрами. Попробуйте применять различные комбинации фильтров к одному и тому же элементу. Вы можете обнаружить интересные визуальные эффекты, которые, кроме того, сделают привлекательными ваши приложения. С помощью фильтров и переходов можно создавать практически те же графические эффекты, что и в графических программных пакетах.
3. Фильтры и переходы могут применяться к элементам блочного уровня, например DIV или P, а к элементам строкового уровня, например таким как STRONG или EM, фильтры и переходы могут применяться только в том случае, если у этого элемента заданы свойства CSS Height или Width.
4. Если фильтр имеет параметры, попробуйте изменять эти параметры и проследите изменения, которые происходят.

Вопросы для самоконтроля:

1. Какие статические фильтры вы знаете?
2. Для чего используют фильтры?
3. К каким элементам страницы можно применять фильтры?
4. Для чего необходимы параметры некоторых фильтров?

5. Как называются динамические фильтры?

4 семестр

Домашняя работа 5. Тема: Язык создания динамических страниц JavaScript.

Задания:

1. Наберите HTML-код для решения простой задачи. Создается документ, в котором при попадании указателя мыши на текст заголовка изменяется цвет шрифта. При перемещении указателя мыши с области заголовка восстанавливается первоначальный цвет шрифта.

```
<html>
<head><title></title></head>
<body bgcolor= silver>
<h3 align = center onmouseout = "this.style.color = 'black' " onmouseover = "this.style.color = 'white' "> Изменение цвета заголовка </h3>
</body>
</html>
```

2. Выполнение пользователем некоторых действий также может приводить к форматированию элементов документа. Наберите HTML – код для решения задачи форматирования элементов документа при выполнении пользователем некоторых действий. Щелчок мыши на абзаце заставит этот абзац разместиться по центру, двойной щелчок заставит абзац выравниваться по правому краю.

```
<html>
<head><title></title></head>
<body bgcolor= silver>
<h3 > Динамическое выравнивание </h3>
<p onClick = "this.style.textAlign = 'center' " onDblClick = "this.style.textAlign = 'right'"> Текст динамически выравнивается </p>
</body>
</html>
```

3. Напишите сценарий, используя элемент <SCRIPT>. Сценарий позволяет отображать или скрывать фрагмент текста. Управление отображением осуществляется двумя кнопками.

```
<html>
<head><title></title>
<script>
function shows(n)
{ var el=document.all["T1"].style
if (n==1)
el.visibility='visible'
else
el.visibility='hidden' }
</script>
</head>
<body bgcolor= silver>
<h3 > Отображение и скрытие текста</h3>
<form name = "form1">
<input type = "button" value = Показать onClick= "shows(1)">
<input type = "button" value = Скрыть onClick = "shows(2)">
</form>
<span id = "T1">
<hr> Текст, который либо показываем, либо скрываем<hr>
</span>
</body>
</html>
```

4. Напишите сценарий, который позволяет скрывать и отображать изображение. Воспользуйтесь для управления отображением свойством `visibility`. Управлять отображением нужно с помощью одной кнопки. На кнопке – надпись с указанием действия, которое можно выполнить. Когда изображение отображено на странице, его скрыть, и наоборот, когда изображение скрыто, его можно отобразить. Каждый раз при выполнении сценария меняется свойство `value` кнопки, поэтому надпись на кнопке соответствует действию, которое можно выполнить применительно к изображению.
5. Управлять отображением документа на странице можно и с помощью свойства `display`. Объект со свойством `display` ведет себя иначе, чем объект со свойством `visibility`. Если значение `display` равно `none`. Элемент не только не отображается на странице, но на странице нет и пустого блока, соответствующего этому элементу. Когда значение `display` равно `block`, элемент отображается на странице.

Вопросы для самоконтроля:

1. Для чего служит тег `<SCRIPT>`?
2. Где располагается в Html-документе скрипт?
3. Какие элементы страницы можно программировать?
4. Что такое пользовательские функции?
5. Как осуществляется вызов функций?

Домашняя работа 6. Тема: Язык создания динамических страниц JavaScript.

Задания:

1. Создайте документ с формой, в которой в текстовые поля пользователь вводит свое имя и оценки за сессию.
2. Необходимо написать программу на JavaScript для подсчета по результатам сессии суммы баллов и среднего балла.
3. Результат выдать в документе в текущем окне в виде простого абзаца.
4. Измените код так, чтобы результат выдавался в таблице в текущем окне:

Студент	введенное имя
Сумма баллов	значение
Средний балл	значение

5. Внесите изменения в код программы так, чтобы та же таблица выдавалась в новом окне.
6. Напишите сценарий, который получает от пользователя три целых числа и выводит в диалоговое окно `alert` сумму, среднее значение, произведение, а так же наибольшее и наименьшее из трех чисел.
7. Измените предыдущий сценарий таким образом, чтобы результаты выводились как текст HTML, как таблица в HTML-документе.

Вопросы для самоконтроля:

1. Является ли JavaScript объектно-ориентированным языком?
2. Что такое коллекции?
3. Назначение объекта `Number`.
4. Перечислите методы объекта `String`.
5. Назначение объекта `Date`.

Домашняя работа 7. Тема: Программирование на стороне сервера с использованием языка PHP.

Задания:

1. Создайте php-скрипт, выводящий страницу сформатированной средствами разметки HTML информацией о вас как о разработчике.
2. Создайте php-скрипт, генерирующий страницу с таблицей основных цветов HTML. Указания: интенсивности красного, зеленого и синего цветов принимают шестнадцатеричные значения 00, 33, 66, 99, CC, FF. Для преобразования между десятичными и шестнадцатеричными числовыми значениями используйте стандартные функции `dechex`, `hexdec`.
3. Реализуйте скрипт, генерирующий и выводящий в браузер случайные числа до тех пор, пока

их сумма не станет больше или равна заданного значения \$n\$. Указание: для генерации псевдо случайного целого числа, принадлежащего диапазону [\$min,\$max], используйте стандартную функцию rand(\$min,\$max).

4. Реализовать ввод и обработку анкеты пользователя. Форма анкеты заполняется на одной странице, скрипт-обработчик, реализованный в отдельном файле, проверяет правильность заполнения всех полей и делает вывод на основе представленной информации (например, вычисляет количество языков программирования, которые знает пользователь, определяет его возраст в годах по введенной дате рождения и т.п.).
5. Реализовать тест из 3-4 вопросов с несколькими вариантами ответа на каждый вопрос, предусмотреть начисление баллов за выбранные пользователем варианты ответа. В конце тестирования, в зависимости от количества набранных баллов, вывести резюме по тесту.
6. Постройте и примените функцию, которая генерирует страницу с HTML-таблицей заданных размеров (количество строк и столбцов). Предусмотрите необязательный параметр функции для закрашивания ячеек таблицы фоновым цветом.
7. Напишите приложение-конвертер, преобразующее одни величины в другие по выбору пользователя, например, температуру по Цельсию в температуру по Фаренгейту, значение, заданное в радианах, в градусы и т.п., предусмотрите 3-4 варианта конвертирования и сохраните информацию о коэффициентах пересчета из одних единиц в другие в массиве.
8. Напишите функцию, производящую вычисление арифметического среднего, несмещенной дисперсии и среднеквадратичного отклонения по выборке, заданной числовым массивом. Результат возвращается в виде массива.
9. Пользователь выполняет ввод произвольного текста в многострочное поле <textarea>. Реализуйте функцию, формирующую отдельные массивы для слов, являющихся и не являющихся допустимой записью целых или вещественных чисел (ненулевых). Функция возвращает два сформированных ей массива.
10. Реализовать web-сервис, позволяющий выполнить следующую обработку введенного или вставленного в многострочное поле текста:
 - замена переводов строк на теги разрыва строки
;
 - замена пустых строк, состоящих только из разделителей, на теги абзаца <p>;
 - удаление пустых строк, состоящих только из разделителей;
 - удаление лишних пробелов или других разделителей между словами;
 - сортировка строк по алфавиту;
 - удаление повторяющихся строк.

Все возможности опциональны и настраиваются через форму ввода данных, учесть, что часть опций могут быть взаимоисключающими.

11. Разработать web-сервис, проверяющий введенные в HTML-форму строки на соответствие выбранному пользователем шаблону. Реализовать (на выбор) следующие шаблоны:
 - быть целым числом;
 - быть вещественным числом;
 - быть идентификатором;
 - быть правильным телефонным номером (например, 237-81-40);
 - быть правильным телефонным номером с кодом города (например, (383) 225-94-00);
 - быть не числом;
 - не содержать цифр;
 - не содержать букв.

Выбор шаблона осуществляется с помощью списка в форме ввода.

12. Реализовать приложение для работы с телефонным справочником, данные которого хранятся в текстовом файле. Предусмотреть возможность сохранения нескольких номеров телефона для одного абонента.
13. Реализовать приложение для проверки доступности сервера, адрес которого введен пользователем в форму.

14. Реализовать приложение для получения и отображения информации, полученной с удаленного сервера (например, прогноза погоды, ленты заголовков новостей).
15. Реализовать приложение файло-обменник, позволяющее посетителям страницы загружать файлы, пока не исчерпан указанный в настройках приложения общий дисковый лимит, а также просматривать список всех загруженных файлов и удалять их. Предусмотреть список разрешенных типов файлов.
16. Реализовать web-приложение, строящее изображение по заданным правилам, например (на выбор):
 - шахматную или иную игровую доску с размещенными на ней фигурами;
 - изображение-мозаику, полученное объединением нескольких заданных изображений;
 - цветовой градиент с гибкими настройками;
 - изображение, содержащее заданный текст и количество дней, оставшееся до выбранной даты;
 - изображение, содержащее информацию о браузере и операционной системе пользователя, которую можно получить средствами PHP.

Вопросы для самоконтроля:

1. Включение PHP-сценария в HTML-документ.
2. Организация ветвлений.
3. Понятие класса. Основные компоненты класса.
4. Абстрагирование, инкапсуляция, модульность и иерархия
5. Хранение и использование данных пользователя. Способы хранения.
6. Хранение данных в файлах. Открытие файла. Функция `fopen()`. Режимы файла. Чтение файла. Запись в файл.
7. Организация счетчика посещений.
8. Обработка форм.
9. Массивы, наследование.

Домашняя работа 8. Тема: Базы данных MySQL.

Задания:

1. Написать программу, которая запрашивает логин пользователя и ищет в базе данных этого пользователя. При положительном результате на web-страницу должна выводиться следующая информация: фамилия и имя пользователя, его возраст, место работы или учебы. При отрицательном результате выводится сообщение, что пользователь не найден.
2. Создайте базу данных с изображениями. Каждому изображению присвойте ключевые слова таким образом, чтобы одно и то же ключевое слово было присвоено разным изображениям. Создайте web-страницу с формой, в поле которой вводится ключевое слово. При нажатии на кнопку "Отобразить" на web-страницу должны выводиться из базы данных все изображения, которым присвоено это ключевое слово.

Вопросы для самоконтроля:

1. Базовая архитектура баз данных для web. Транзакция базы данных для web.
2. Выбор базы данных.
3. Структура процедуры доступа. Установка соединения.
4. Фильтрация входных данных.
5. Выполнение запроса к базе данных.
6. Получение результатов запроса.
7. Организация поиска в базе данных.

Примерный вариант теста по теме «Язык гипертекстовой разметки страниц HTML»

1. Требуется написать химическую формулу Сульфида мышьяка (As_2S_3). Каким образом это можно сделать?
 - a) `As<sup>2S<sup>3`
 - b) `As₂S₃`

- c) $As^{2}S^{3}$
d) $As^{2}S^{3}$
2. В каких случаях атрибут выравнивания align имеет более высокий приоритет?
- `<TH align="left">`
 - `<COL align="left">`
 - `<TABLE align="left">`
 - `<OL align="left">`
 - `<UL align="left">`
3. Какой вариант написан с ошибкой?
- `<ol type="N">`
 - `<ol type="I">`
 - `<ol type="i">`
 - `<ol type="A">`
4. Необходимо сделать ширину таблицы на всю страницу (либо родительского контейнера). Как надо написать тег `<table>`?
- `<table>`
 - `<table width="auto">`
 - `<table width="100%">`**
 - `<table width="100">`
5. Какой из приведенных фрагментов кода создает переключатель?
- `<input Type="checkbox" nAME="a1" vAlue="1"><input TYPE="checkbox" nAME="a1" vAlue="2"><input TYpe="text" nAME="a1" vAlue="2">`
 - `<input TYPE="radiobutton" nAME="a1" vAlue="1"><input TYPE="radiobutton" nAME="a1" vAlue="2">`
 - `<input TYPE="radio" nAME="a1" vAlue="1"><input TYPE="radio" nAME="a1" vAlue="2">`**
 - `<input Type="checkbox" nAME="a1" vAlue="1"><input TYPE="checkbox" nAME="a1" vAlue="2"><input TYpe="text" nAME="a1" vAlue="2">`
 - `<input TYPE="radiobutton" nAME="a1" vAlue="1"><input TYPE="radiobutton" nAME="a1" vAlue="2">`
6. На сайте внутри папки pages находится файл page.html. А внутри папки images находится файл foto.jpg. Причём папки images и pages лежат в корне сайта. Как правильно написать путь к foto.jpg из файла page.html:
- `../images/foto.jpg`**
 - `../images/pages/foto.jpg`
 - `pages/images/foto.jpg`
 - `images/foto.jpg`
7. Какого тега НЕ существует:
- `<pre>`
 - ``
 - `<adress>`**
 - ``
8. Какой атрибут элемента FORM определяет список кодировок для вводимых данных?
- alt
 - accept-charset**
 - enctype-charset
 - act-charset
 - enct-charset
9. Какой из вариантов содержит ошибку:
- `Ссылка`
 - `Ссылка`
 - `Ссылка`**

- d) `Ссылка`
10. Размер окна браузера 1000 пикселей. На страницу добавили блок с шириной 40%. Затем в этот блок добавили таблицу с шириной 50%. Какова будет ширина таблицы в пикселях?
- a) 250 пикселей.
 - b) Недостаточно данных.
 - c) **200 пикселей.**
 - d) 500 пикселей.

Примерный вариант теста по теме «Использование стиля при оформлении сайта. Спецификации CSS»

1. Как изменить цвет фона для всех элементов h1 на странице?
 - a) **h1 {background-color: #ccc;}**
 - b) `h1[all] {background-color: #ccc;}`
 - c) `h1.all {background-color: #ccc;}`
 - d) `h1:all {background-color: #ccc;}`
2. Написан HTML-код: `<div id="myid">Содержимое</div>`. Как задать стиль для тега `<div>`?
 - a) `div.myid {margin: 1px;}`
 - b) `.myid {margin: 1px;}`
 - c) **`div#myid {margin: 1px;}`**
 - d) `div[myid] {margin: 1px;}`
3. Какой вариант задания цвета НЕ сработает?
 - a) `color: #aaa;`
 - b) `color: #aaaaaa;`
 - c) `color: #000;`
 - d) **`color: #hhh;`**
4. Как правильно вставляются комментарии в CSS-код?
 - a) **`/* Мой комментарий */`**
 - b) `# Мой комментарий #`
 - c) `# Мой комментарий`
 - d) `// Мой комментарий`
5. Написан HTML-код: `<p>Синий не синий</p>`. Какой CSS-код внутри тега `` сделает синий цвет?
 - a) `span {color: blue;}`
 - b) `p span {color: blue;}`
 - c) **Все представленные варианты подойдут**
 - d) `.blue {color: blue;}`
6. Шрифт задан следующим образом: `font-family: Arial, Times New Roman, Helvetica, sans-serif;`. Что здесь не так?
 - a) **Нужно поставить кавычки у Times New Roman.**
 - b) Шрифта Helvetica не существует.
 - c) Нельзя указывать больше 3-х различных шрифтов.
 - d) Вместо свойства `font-family` нужно использовать свойство `font`.
7. Какой селектор написан с ошибкой?
 - a) **`input["text"]`**
 - b) `input[type="text"]`
 - c) `input.text`
 - d) `div p`
8. Какое принципиальное отличие селектора ID от селектора CLASS?
 - a) Никакого отличия между ними нет.
 - b) **ID должен быть уникальным на странице, а одинаковый CLASS может быть у нескольких элементов.**

- c) Разница только в наборе свойств, который может быть использован для этих селекторов. Например, для ID нельзя задавать свойство border, а для CLASS можно.
 - d) CLASS должен быть уникальным на странице, а одинаковый ID может быть у нескольких элементов.
9. Есть такой CSS-код во внешнем файле: `p {color: blue;}`. На странице написан такой HTML-код: `<p style="color: red;">текст</p>`. Какого цвета будет "текст"?
- a) Синего.
 - b) Чёрного.
 - c) В браузере IE8 синего, а в других красного.
 - d) Красного.**
10. Необходимо, чтобы при наведении указателя мыши на ссылку, она меняла цвет. Какой псевдоэлемент нужно использовать?
- a) active
 - b) onmouseover
 - c) link
 - d) hover**

Примерный вариант теста по теме «Язык создания динамических страниц JavaScript»

1. Написан следующий код:

```
<script type="text/javascript">
  var a = 1;
  function func() {
    a = 10;
  }
  func();
  alert(a);
</script>
```

Что будет выведено в появившемся окне?

- a) 1
 - b) undefined
 - c) 10**
 - d) Ошибка, поскольку в коде идёт попытка 2 раза объявить переменную с одинаковым именем.
2. Какая разница между равенством и эквивалентностью?
- a) Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.
 - b) Никакой разницы нет.
 - c) Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.
 - d) Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.**
3. Написан следующий код:

```
<script type="text/javascript">
  var a = 1;
  var b = a++ + 1;
  alert(b);
</script>
```

Что будет выведено в появившемся окне?

- a) Ничего не будет, поскольку инкремент нельзя соединять со знаком сложения.
 - b) 1
 - c) 3
 - d) 2**
4. Что делает функция `printMessage("Text")`?
- a) Ошибка, поскольку в функции `printMessage()` 2 параметра.
 - b) Выводит окно с текстом Text.

c) Функция `printMessage()` вообще не существует.

d) Выводит окно с текстом "Text".

5. Написан следующий код:

```
<script type="text/javascript">
  var a = 1;
  function func() {
    var a = 10;
  }
  func();
  alert(a);
</script>
```

Что будет выведено в появившемся окне?

a) 1

b) Ошибка, поскольку в коде идёт попытка 2 раза объявить переменную с одинаковым именем.

c) undefined

d) 10

6. Какая функция вызывает окно с текстовым полем, в которое пользователь может ввести строку?

a) `alerts()`

b) **`prompt()`**

c) `promt()`

d) `alert()`

7. Написан следующий код:

```
<script type="text/javascript">
  var arr = new Array();
  arr[arr.length] = 0;
  arr[arr.length] = 1;
  alert(arr.length);
</script>
```

Что будет выведено в появившемся окне?

a) 0

b) 2

c) 1

d) Код нерабочий, поскольку писать `arr[arr.length]` можно только при длине массива отличной от 0.

8. Есть код:

```
<script type="text/javascript">
  var a = 1;
  function func() {
    alert(a);
  }
  func();
</script>
```

Что будет написано в появившемся окне?

a) Ничего не будет, поскольку возникнет ошибка "переменная не определена".

b) undefined

c) 0

d) 1

9. Вызваны две функции: `parseFloat("$101.2")` и `parseFloat("101.2 USD")`. Какая из них вернёт число?

a) `parseFloat("$101.2")`

b) **`parseFloat("101.2 USD")`**

c) Обе функции.

d) Ни одна из них.

10. Есть код:

```
<script type="text/javascript">
  var a = "5";
  var b = 2;
  var c = a + b;
  alert(c);
</script>
```

Что будет написано в появившемся окне?

- a) NaN
- b) 7
- c) Ошибка выполнения скрипта.
- d) 52

Примерный вариант теста по теме «Программирование на стороне сервера с использованием языка PHP»

1. Есть строка: "my string". Есть код:

```
<?php
  if (strpos("mystring", "m") == false) echo "Символа m в строке mystring нет";
?>
```

В чём заключается ошибка в коде?

- a) Здесь нет ошибок, так как strpos("mystring", "m") не равен false, и строки "Символа m в строке mystring нет" не появляется.
 - b) Функция strpos() неправильно использована. Сначала должна идти искомая строка, а уже потом та строка, в которой происходит поиск.
 - c) **Функция strpos() вернула 0, который в PHP равен false. Чтобы не было ошибки надо вместо знака равенства (==) использовать знак эквивалентности (===).**
 - d) Нет фигурных скобок у блока операторов при срабатывании условия.
2. Есть массив \$arr. Как вывести все элементы массива \$arr?
- a) echo \$arr;
 - b) Это возможно сделать только через цикл.
 - c) print \$arr;
 - d) **print_r(\$arr);**
3. Каков результат выполнения у данного скрипта:

```
<?php
  $x = 5;
  echo "Переменная $x = 5";
?>
```

- a) Ошибка, поскольку строки с переменными должны быть всегда в одинарных кавычках.
 - b) **Переменная 5 = 5**
 - c) Переменная \$x = 5
 - d) Переменная x = 5
4. Какая разница между равенством и эквивалентностью?
- a) Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.
 - b) **Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.**
 - c) Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.
 - d) Никакой разницы нет.
5. Каков результат выполнения у данного скрипта:

```
<?php
  for ($i = 0; $i < 5; $i++) {
    if ($i % 2 == 0) continue;
    echo $i;
  }
?>
```

- a) 013

- b) 024
- c) 24
- d) 13**

6. Какого оператора в PHP не существует?

- a) != =**
- b) != =
- c) +=
- d) >=

7. Что выведет такой код:

```
<?php
$x = 10;
$x += --$x;
echo $x;
?>
```

- a) 19
- b) 20
- c) 18**
- d) 21

8. Каков будет результат выполнения скрипта:

```
<?php
function myfunc($x = 0) {
    echo $x;
}
myfunc();
myfunc(5);
?>
```

- a) Ошибка, поскольку параметрам в функции нельзя присваивать значения.
- b) 0
- c) 5
- d) 05**

9. Как сделать редирект (например, на google.ru) на PHP?

- a) location.href = "http://google.ru";
- b) header("Redirect: http://google.ru");
- c) header("Location: http://google.ru");**
- d) document.location = "http://google.ru";

10. Какого типа данных в PHP нет?

- a) Resource.
- b) null.
- c) Объект.
- d) Указатель.**

Примерный вариант теста по теме «Базы данных MySQL»

1. Какое из утверждений о первичном ключе НЕ верно?

- a) Первичный ключ может содержать NULL значения.**
- b) Каждая таблица имеет первичный ключ.
- c) Первичный ключ содержит только уникальные значения.
- d) Первичный ключ НЕ может содержать NULL значений.

2. В некоторой таблице есть поле "field". Как правильно вывести все записи, где значение поля "field" содержит строку (либо подстроку) "string":

- a) WHERE `field` LIKE "string"
- b) WHERE `field` == "string"
- c) WHERE `field` LIKE "%string%"**
- d) WHERE `field` = "string"

3. Как удалить таблицу "my_table":

- a) DELETE FROM "my_table"
 - b) DELETE TABLE "my_table"
 - c) **DROP TABLE "my_table"**
 - d) DELETE "my_table"
4. Выберите верное утверждение относительно индекса:
- a) Индекс ускоряет добавление записей в таблицу.
 - b) **Индекс позволяет ускорить выборку с тем полем, для которого он сделан.**
 - c) Индекс ускоряет абсолютно любые запросы с таблицей.
 - d) Индекс позволяет сэкономить место, занимаемое таблицей.
5. Для чего используется команда GRANT?
- a) Для очистки таблицы.
 - b) **Для управления пользователями.**
 - c) Для перезапуска сервера MySQL.
 - d) Для очистки базы данных.
6. Написан такой запрос: SELECT text FROM articles WHERE id='15'
Однако он не срабатывает. Почему?
- a) Таблицу articles надо заключить в обратные кавычки, поскольку это зарезервированное слово.
 - b) **Поле text надо заключить в обратные кавычки, поскольку это зарезервированное слово.**
 - c) Поле id надо заключить в обратные кавычки, поскольку это зарезервированное слово.
 - d) Надо писать не id='15', а id="15".
7. Для чего используется ключевое слово DISTINCT?
- a) Для снижения нагрузки на сервер с потерей производительности выполнения запроса.
 - b) **Для выборки только уникальных записей по конкретному полю.**
 - c) Для ускорения выборки по конкретному полю.
 - d) Такого ключевого слова не существует.
8. Необходимо извлечь все записи, где значение поля "field" начинается с подстроки "abc".
Какой запрос для этого надо использовать?
- a) SELECT * FROM `my_table` WHERE `field` LIKE '%abc%'
 - b) SELECT * FROM `my_table` WHERE `field` <> 'abc'
 - c) SELECT * FROM `my_table` WHERE `field` STARTSWITH 'abc'
 - d) **SELECT * FROM `my_table` WHERE `field` LIKE 'abc%'**
9. Что делает конструкция: ORDER BY ASC
- a) Сортирует записи по убыванию.
 - b) ASC вообще не существует.
 - c) ASC - это функция, которая здесь некорректно использована.
 - d) **Сортирует записи по возрастанию.**
10. Как обновить запись сразу в двух колонках:
- a) **UPDATE `my_table` SET `field_1` = 'value_1', `field_2` = 'value_2'**
 - b) UPDATE `my_table` SET `field_1` = 'value_1' SET `field_2` = 'value_2'
 - c) UPDATE `my_table` SET `field_1` = 'value_1', SET `field_2` = 'value_2'
 - d) UPDATE `my_table` SET `field_1` = 'value_1' AND `field_2` = 'value_2'

Примерные вопросы к экзамену (3 семестр):

- 1. Глобальные компьютерные сети: основные понятия, принципы функционирования.
- 2. Каталоги ресурсов. Поисковые системы.
- 3. Язык гипертекстовой разметки страниц HTML: общая структура документа, абзацы, цвета, ссылки.
- 4. Язык гипертекстовой разметки страниц HTML: списки, графика (графические форматы, графический объект как ссылка).

5. Язык гипертекстовой разметки страниц HTML: таблицы.
6. Язык гипертекстовой разметки страниц HTML: формы.
7. Фреймы.
8. Общие подходы к дизайну сайта. Разработка макета страницы
9. Использование стиля при оформлении сайта. Возможности CSS.
10. CSS. Свойства текста. Свойства цвета и фона.
11. CSS. Свойства шрифта. Свойства блоков.
12. CSS. Свойства списков. Классы. Псевдоклассы.

Примерные вопросы к экзамену (4 семестр):

1. Хостинг. Бесплатный хостинг. FTP. Размещение Интернет-ресурса на сервере провайдера.
2. Регистрация Интернет-ресурса в каталогах и поисковых системах.
3. Преимущества и ограничения программ, работающих на стороне клиента. Язык JavaScript: основы синтаксиса.
4. Введение в программирование на стороне сервера на примере PHP. Принцип работы.
5. Синтаксис языка программирования PHP.
6. Переменные. Константы. Операторы в PHP. Циклы. Массивы. Работа со строками.
7. Функции в PHP. Встроенные функции.
8. Работа с датой и временем в PHP.
9. Методы передачи параметров между страницами (GET, POST). Обработка действий пользователя при помощи форм.
10. Принципы хранения информации в базах данных MySQL. Архитектура базы данных MySQL (таблицы, связи, триггеры).
11. Подключение к базе данных из PHP файла. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу. Передача параметров в запрос.
12. Принципы проектирования страниц. Разделение информации по таблицам в базе данных. Вывод группы данных, сортировка данных.
13. Создание HTML-страниц средствами PHP.

Критерии оценивания (экзамен)

Оценка «отлично» выставляется студенту, глубоко и прочно усвоившему программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагающему, в ответе которого увязывается теория с практикой, он показывает знакомство с литературой, правильно обосновывает и использует рациональные и современные средства решения поставленной проблемы.


Оценка «хорошо» выставляется студенту, твердо знающему программный материал, грамотно и по существу излагающему его, который не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении поставленной задачи.

Оценка «удовлетворительно» выставляется студенту, который знает только основной программный материал, но не усвоил особенностей, допускает в ответе неточности, некорректно формулирует основные законы и правила, затрудняется в выполнении практических задач.

Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает в ответе существенные ошибки, с затруднениями выполняет практические задания.

Форма контроля	За одну работу		Всего	
	Мин. баллов	Макс. баллов	Мин. баллов	Макс. баллов
Текущий контроль:				
Активная работа на занятии	0,25	0,5	15	18
Выполнение домашнего задания	0,13	0,25	4,5	9
Выполнение заданий лабораторной работы	0,25	0,5	15	18

Выполнение заданий самостоятельной работы	1	3	1	3
Выполнение теста	0,3	0,6	1,5	3
Ответ на коллоквиуме	3	6	3	6
Промежуточная аттестация (экзамен)			20	43
Итого за 5 семестр /экзамен			60	100

Составитель _____  _____ Соболев Е.И.,
руководитель группы сопровождения хостинговой платформы ГКУ Сахалинской области «Центр
Региональной Цифровой Трансформации»

«12 » марта 2024 г