



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Филиал федерального государственного бюджетного образовательного учреждения
высшего образования «Кубанский государственный университет»
в г. Славянске-на-Кубани



УТВЕРЖДАЮ

Проректор по работе с филиалами

А.А. Евдокимов

31 августа 2016 г.

Рабочая программа учебной дисциплины

ОП.12 WEB-ПРОГРАММИРОВАНИЕ

специальность 09.02.02 Компьютерные сети


Рабочая программа учебной дисциплины ОП.12 «Web-программирование» разработана на основе Федерального государственного образовательного стандарта (далее — ФГОС) по специальности среднего профессионального образования (далее СПО) 09.02.02 Компьютерные сети, утвержденного приказом Минобрнауки РФ от 28.07.2014 №803 (зарегистрирован в Минюсте России 20.08.2014 № 33713).

Дисциплина	ОП.12 «Web-программирование»
Форма обучения	Очная
Учебный год	2016-2017
2, 3 курс	7 семестр
Лекции	20 час.
практические занятия	30 час.
лабораторные занятия	—
самостоятельные занятия	25 час.
форма промежуточного контроля	Диффер. зачет

Составитель: преподаватель  А. Н. Чернышев, Трепалин И.О.
подпись

Утверждена на заседании предметно-цикловой комиссии физико-математических дисциплин и специальных дисциплин специальности Компьютерные сети, протокол №1 от 29 августа 2016 г.

Председатель предметно-цикловой комиссии физико-математических дисциплин и специальных дисциплин специальности Компьютерные сети

 А. Б. Шишкин
29 августа 2016 г.

Рецензент (-ы):


Инженер -программист 1 категории, отдел УСУТП управление АСУТП, КИПиА, МОП Краснодарского РПУ филиала «Макрорегион ЮГ» ООО ИК «Сибинтек»		Литус М.В.
начальник отдела информационных технологий ОАО «Сад-Гигант»		Дудник П.А.

ЛИСТ
согласования рабочей программы по учебной дисциплине
ОП.12 «Web-программирование»

Специальность среднего профессионального образования:
09.02.02 Компьютерные сети


СОГЛАСОВАНО:

Нач. УМО филиала




А. В. Баранов
29 августа 2016 г.

Заведующая библиотекой филиала



М. В. Фуфалько
29 августа 2016 г.

Начальник ИВЦ (программно-информационное
обеспечение образовательной программы)



В. А. Ткаченко
29 августа 2016 г.

Содержание

1. Паспорт рабочей программы учебной дисциплины	6
1.1. Область применения программы	6
1.2. Место дисциплины в структуре программы подготовки специалистов среднего звена	6
1.3. Цели и задачи дисциплины, требования к результатам освоения дисциплины....	6
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)	7
2. Структура и содержание учебной дисциплины	7
2.1. Объем учебной дисциплины и виды учебной работы	7
2.2. Структура дисциплины	7
2.3. Тематический план и содержание учебной дисциплины	8
2.4. Содержание разделов дисциплины	9
2.4.1. Занятия лекционного типа	9
2.4.2. Практические занятия	12
2.4.3. Лабораторные занятия	13
2.4.4. Примерная тематика курсовых работ.....	13
2.4.5. Содержание самостоятельной работы.....	13
2.4.6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	15
3. Образовательные технологии	17
3.1. Образовательные технологии при проведении лекций	17
3.2. Образовательные технологии при проведении практических занятий.....	18
3.3. Образовательные технологии при проведении лабораторных занятий	18
4. Условия реализации программы учебной дисциплины	19
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	19
4.2. Перечень необходимого программного обеспечения	19
5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	20
5.1 Основная литература	20
5.2 Дополнительная литература	20
5.3 Периодические издания	20
5.4 Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины	21
6. Методические указания обучающимся по освоению дисциплины	22
7. Оценочные средства для контроля успеваемости.....	26
7.1. Паспорт фонда оценочных средств.....	26
7.2. Критерии оценки результатов обучения	26
7.3. Оценочные средства для проведения текущей аттестации	27
7.4. Оценочные средства для проведения промежуточной аттестации	32
7.4.1. Примерные вопросы для проведения промежуточной аттестации	32

7.4.2. Примерные задачи для проведения промежуточной аттестации	33
8. Дополнительное обеспечение дисциплины.....	33

1. Паспорт рабочей программы учебной дисциплины ОП.12 «Web-программирование»

1.1. Область применения программы

Рабочая программа учебной дисциплины ОП.12 «Web-программирование» является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.02 «Компьютерные сети».

1.2. Место дисциплины в структуре программы подготовки специалистов среднего звена

Дисциплина входит в профессиональный цикл, в раздел общепрофессиональных дисциплин.

При изучении дисциплины используются знания и умения, сформированные в процессе изучения дисциплин ОП.04 «Операционные системы», ОП.05 «Основы программирования и баз данных», МДК.02.01 «Программное обеспечение компьютерных сетей», МДК.02.02 «Организация администрирования компьютерных систем». Успешное изучение дисциплины способствует углубления знаний и умений, полученных в процессе изучения дисциплин ОП.05 «Основы программирования и баз данных», МДК.02.01 «Программное обеспечение компьютерных сетей», МДК.02.02 «Организация администрирования компьютерных систем».

1.3. Цели и задачи дисциплины, требования к результатам освоения дисциплины

В результате изучения профессионального модуля обучающийся должен:

знать:

- методы проектирования web-сайта как статичной информационной системы;
- методы проектирования web-сайта как динамичной информационной системы;
- теорию использования графики на web-страницах;
- методы обработки и редактирования цифровых изображений;
- программные средства стороны клиента, используемые для создания web-страниц;
- программные средства стороны сервера, используемые для создания web-страниц;
- программные средства для создания баз данных;
- программные средства создания web-сервера;
- основные принципы конфигурации реального web-сервера;
- программные средства, используемые для размещения и сопровождения web-страниц;

уметь:

- использовать графические редакторы для обработки изображений, размещаемых на web-сайте;
- использовать язык гипертекстовой разметки HTML для создания web-страниц;
- создавать динамические web-страницы с использованием JavaScript;
- использовать PHP для создания web-страниц;
- осуществлять доступ к базам данных при проектировании web-сайта;
- настраивать конфигурацию web-сервера;

иметь практический опыт:

- проектирования web-сайта;
- оптимизации изображений для размещения на web-сайте;
- создания web-сайта средствами программирования на стороне клиента;
- проектирования web-сайта на стороне сервера;
- создания баз данных на стороне сервера;
- размещения web-сайта на сервере.

Максимальная учебная нагрузка обучающегося в 7 семестре 75 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 50 часов;
- самостоятельная работа обучающегося 25 часов.

1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)

Учащийся должен обладать общими компетенциями, включающими в себя способность:

- ПК 2.2. Администрировать сетевые ресурсы в информационных системах.
- ПК 2.3. Обеспечивать сбор данных для анализа использования и функционирования программно-технических средств компьютерных сетей.
- ПК 3.1. Устанавливать, настраивать, эксплуатировать и обслуживать технические и программно-аппаратные средства компьютерных сетей.

Структура и содержание учебной дисциплины

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестры
		7
Обязательная учебная нагрузка (всего)	50	50
В том числе:		
занятия лекционного типа	20	20
практические занятия (практикумы)	30	30
лабораторные занятия	–	–
Самостоятельная работа (всего)	25	25
в том числе:		
<i>Реферат</i>	10	10
<i>Самостоятельная внеаудиторная работа в виде домашних практических заданий, индивидуальных заданий, самостоятельного подбора и изучения дополнительного теоретического материала и др.</i>	15	15
Вид промежуточной аттестации		Диффер. зачет
Общая трудоемкость	75	75

2.2. Структура дисциплины

Учебная дисциплина ОП.12 «Web-программирование» имеет следующую структуру:

	Всего	Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа
Раздел 1. HTML и CSS. Основы создания сайта.	18	4	4	-	10
Раздел 2. Клиентское web-программирование. JavaScript.	17	6	6	-	5
Раздел 3. Серверное web-программирование. PHP и MySQL.	40	10	20	-	10

2.3. Тематический план и содержание учебной дисциплины ОП.12 «Web-программирование»

Наименование разделов и тем	Содержание учебного материала, практические работы, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
Раздел 1. HTML и CSS. Основы создания сайта.	Содержание учебного материала		
	Лекции	4	
	Лекция 1. Основы HTML. Каскадные таблицы стилей (CSS).	2	1, 2
	Лекция 2. Использование таблиц для верстки. Использование слоев в HTML. Верстка при помощи слоев.	2	1, 2
	Практические занятия	4	
	Практическое занятие 1. Основы HTML. Каскадные таблицы стилей (CSS).	2	
	Практическое занятие 2. Использование таблиц для верстки. Верстка при помощи слоев.	2	
	Лабораторные занятия	–	
	Самостоятельная работа обучающихся	10	
Раздел 2. Клиентское web-программирование. JavaScript.	Содержание учебного материала		
	Лекции	6	
	Лекция 3. Введение в JavaScript. Способы внедрения JS, стандартные функции. Типы переменных. Работа с переменными и операторами сравнения.	2	1, 2
	Лекция 4. Управление web-объектами. Знакомство с событиями. Работа с формами, валидация.	2	1, 2
	Лекция 5. Классы и прототипы. Cookies.	2	1, 2
	Практические занятия	6	
	Практическое занятие 3. Способы внедрения JS, стандартные функции. Работа с переменными и операторами сравнения.	2	
	Практическое занятие 4. Управление web-объектами. Знакомство с событиями. Работа с формами, валидация.	2	
	Практическое занятие 5. Работа над проектом «Разработка сайта средствами HTML и JS».	2	
	Лабораторные занятия	–	
	Самостоятельная работа обучающихся	5	
Семестр 7			
Раздел 3. Серверное web-программирование. PHP и MySQL.	Содержание учебного материала		
	Лекции	10	
	Лекция 6. Основы PHP. Условные операторы. Понятие функции.	2	1, 2
	Лекция 7. Циклы и массивы. Запросы HTTP, параметры URL и формы HTML.	2	1, 2
	Лекция 8. Стандартные средства PHP. Использование файлов в PHP.	2	1, 2
	Лекция 9. СУБД MySQL. Разработка базы данных web-приложения. Взаимодействие PHP и MySQL.	2	1, 2
	Лекция 10. Авторизация пользователей. Сессии. Использование Cookie. Почтовые рассылки в PHP.	2	1, 2
	Практические занятия	20	
	Практическое занятие 6. Основы PHP. Условные операторы. Понятие функции.	2	
	Практическое занятие 7. Циклы и массивы.	2	
	Практическое занятие 8. Запросы HTTP, параметры URL и формы HTML.	2	
	Практическое занятие 9. Стандартные средства PHP.	2	
	Практическое занятие 10. Использование файлов в PHP.	2	
	Практическое занятие 11. СУБД MySQL. Разработка базы данных web-приложения.	2	
	Практическое занятие 12. Взаимодействие PHP и MySQL.	2	
	Практическое занятие 13. Авторизация пользователей. Сессии.	2	

Наименование разделов и тем	Содержание учебного материала, практические работы, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
	Практическое занятие 14. Использование Cookie. Почтовые рассылки в PHP.	2	
	Практическое занятие 15. Работа над проектом «Расширение функциональности сайта средствами PHP и MySQL».	2	
	Лабораторные занятия	–	
	Самостоятельная работа обучающихся	10	

Уровень освоения: 1 - легкий, 2 - относительно легкий, 3 — сложный.

2.4. Содержание разделов дисциплины

2.4.1. Занятия лекционного типа

№ раздела	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
<i>6 семестр</i>			
Раздел 1. HTML и CSS. Основы создания сайта.			
1	Основы HTML.	<ul style="list-style-type: none"> • Структура HTML страницы: заголовок и тело документа. • Использование в html комментариев и спецсимволов. • Теги для оформления текста. • Работа со стилем текста, его разбитие на блоки. • Гиперссылки, их объявление на странице. • Иллюстрации на веб-странице и их правильное использование. • Обмен с сервером. Формы и элементы управления. 	Р, У
2	Каскадные таблицы стилей (CSS).	<ul style="list-style-type: none"> • Понятие и назначение CSS. • Синтаксис CSS. • Селекторы (id, class, tag). • Создание стилей и задание их свойств. • Вложенность, наследование и группы свойств. • Приоритеты применения стилей. • Псевдоклассы и псевдоэлементы. 	Р, У
3	Использование таблиц для верстки.	<ul style="list-style-type: none"> • Теги HTML для создания таблиц. • Свойства таблицы и их использование для верстки сайтов. • Задание шаблона сайта при помощи таблиц. • Объединение ячеек. • Вложенные таблицы. • Использование атрибутов отображения таблиц. • Табличная верстка. 	Р, У
4	Использование слоев в HTML. Верстка при помощи слоев.	<ul style="list-style-type: none"> • Слои и их использование. Верстка слоями. • Методика верстки. Теги (div и span). • Использование атрибутов тегов. Отступы элементов (margin и padding). • Размещение блоков на странице. • Заготовка для разметки сайта. • Применение overflow и clear в реальной вёрстке. • Заполнение разметки частями нарезанного макета. • Методы размещения элементов при помощи слоев. 	Р, У
Раздел 2. Клиентское web-программирование. JavaScript.			

№ разд ела	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
5	Введение в JavaScript. Способы внедрения JS, стандартные функции.	<ul style="list-style-type: none"> • История JavaScript. • Понятие JavaScript. • Область применения JavaScript. • Литералы, переменные, выражения. • JavaScript-код на странице. • Подключение внешнего JavaScript файла. • Знакомство с alert() и console(). • Самые распространенные ошибки в JavaScript. 	Р, У
6	Типы переменных. Работа с переменными и операторами сравнения.	<ul style="list-style-type: none"> • Числа. • Строки. • Массивы. • Регулярные выражения. • Простые математические операции. • Использование оператора if & else. • Оператор switch. • Обработка ошибок Try & catch. • Знакомство с JavaScript-методами: Math, String, Window. 	Р, У
7	Управление web-объектами. Знакомство с событиями.	<ul style="list-style-type: none"> • Знакомство с функциями. • Знакомство с событиями. • Событие CLICK. • Событие MouseOver. • Работа с окнами. 	Р, У
8	Работа с формами, валидация.	<ul style="list-style-type: none"> • Получение параметра ввода пользователя. • Изменения параметра ввода, преобразование. • Валидация. 	Р, У
9	Классы и прототипы.	<ul style="list-style-type: none"> • Конструкторы. • Прототип и наследование свойств. • Создание класса, обращение к методам класса, удаление класса. 	Р, У
10	Cookie.	<ul style="list-style-type: none"> • История cookie. • Безопасность и ограничения. • Область применения cookie. • Чтение cookie. Удаление cookie. 	Р, У
Раздел 3. Серверное web-программирование. PHP и MySQL.			
11	Основы PHP.	<ul style="list-style-type: none"> • История и назначение PHP. Особенности языка. • Установка и настройка связки Apache, PHP, MySQL. Сборка Denwer. • Синтаксис языка PHP. • Пишем «Hello, World». • Переменные в языке PHP. Типы данных. • Преобразование типов данных. • Константы. • Арифметические операции. Интерпретация арифметических операций для строк. • Логические операции. Интерпретация логических операций для строк. 	Р, У

№ разд ела	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
12	Условные операторы. Понятие функции.	<ul style="list-style-type: none"> • Условия при выполнении программы. Ветвление алгоритма. • Оператор if. • Оператор switch. • Функции в языке PHP. Использование функций. • Объявление и определение функции. • Передаваемые в функцию параметры. Задание значения по умолчанию для передаваемых параметров. • Возвращение значения из функции. Оператор return. • Рекурсивный вызов функции. Применение рекурсии. • Область видимости и время жизни переменных в программе на PHP. 	Р, У
13	Циклы и массивы.	<ul style="list-style-type: none"> • Использование циклов в PHP. • Циклы while и do..while. • Циклы for и foreach. • Организация бесконечного цикла. Бесконечный цикл как наиболее распространенная ошибка начинающего программиста. • Оператор выхода из цикла. • Оператор прерывания итерации цикла. • Массивы. Объявление массивов в языке PHP. • Использование циклов для работы с массивом. • Функции для работы с массивами. • Многомерные массивы: отличие от одномерных и работа с ними в PHP. • Предопределенные массивы. 	Р, У
14	Запросы HTTP, параметры URL и формы HTML.	<ul style="list-style-type: none"> • Принципы работ протокола HTTP. • Запросы GET и POST. • Метод GET. Передача пользовательских параметров через URL. Параметры запроса в URL. • Обработка параметров, полученных методом GET в PHP. • Отправка параметров методом POST. Обработка отправки запросов через HTML форму. • Преимущества и недостатки методов GET и POST. 	Р, У
15	Стандартные средства PHP.	<ul style="list-style-type: none"> • Документация по PHP в сети Internet. • Встроенные функции языка PHP. • Вычисление математических функций. • Обработка строк. Регулярные выражения. • Функции для работы с массивами. • Работа с переменными даты и времени. • Встроенные константы. • Переменные среды. • Функции эмуляции SSI. 	Р, У
16	Использование файлов в PHP.	<ul style="list-style-type: none"> • Принципы работы с файлами в PHP. • Режимы открытия файла. Чтение и запись. • Средства языка для чтения и записи файлов. • Разработка модуля записи данных пользователя в файл. • Обработка ошибок при работе с файлами. • Организация загрузки файла на сайт. • HTML форма, обеспечивающая загрузку. Обработка загруженных файлов языком PHP. • Работы с каталогами, расположенными на сервере. • Получение списка файлов и директорий. 	Р, У

№ разд ела	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
17	СУБД MySQL. Разработка базы данных web- приложения.	<ul style="list-style-type: none"> Использование СУБД MySQL для веб-разработки. Установка и конфигурация MySQL под ОС Windows. Программы для работы с сервером баз данных MySQL. Установка PhpMyAdmin. Использование PhpMyAdmin для разработки базы данных. Экспорт и импорт баз данных. Проектирование базы данных. Первичные и вторичные ключи. Графическое представление базы данных. ER-диаграмма. 	Р, У
18	Взаимодействие PHP и MySQL.	<ul style="list-style-type: none"> Расширения языка PHP для MySQL. Соединение с сервером баз данных. Запросы для получения информации. Обработка результата запроса. Обеспечение безопасности данных. Борьба с SQL-инъекциями. 	Р, У
19	Авторизация пользователей. Сессии.	<ul style="list-style-type: none"> Использование сессий. Создание сессии. Хранение данных о пользователе. Хранение паролей в БД. Хеширование и алгоритм md5. Форма регистрации пользователя. Сохранение профиля. Авторизация пользователя. Чтение данных сессии. Выход пользователя с сайта. Закрытие сессии. 	Р, У
20	Использование Cookie. Почтовые рассылки в PHP.	<ul style="list-style-type: none"> Хранение пользовательских данных в Cookie. Что можно хранить в cookie. Типы cookie: постоянные, временные. Работы с cookie: создание, чтение, удаление. Как работает электронная почта. Расширения языка PHP для работы с электронной почтой. Создание электронного письма средствами PHP. 	Р, У
Примечание: Т – тестирование, Р – написание реферата, У – устный опрос			

2.4.2. Практические занятия

№	Наименование раздела	Наименование практических работ	Форма текущего контроля
1	2	3	4
1	Раздел 1. HTML и CSS. Основы создания сайта.	Практическое занятие 1. Основы HTML. Каскадные таблицы стилей (CSS). Практическое занятие 2. Использование таблиц для верстки. Верстка при помощи слоев.	ПР, У
2	Раздел 2. Клиентское web- программирование. JavaScript.	Практическое занятие 3. Способы внедрения JS, стандартные функции. Работа с переменными и операторами сравнения. Практическое занятие 4. Управление web-объектами. Знакомство с событиями. Работа с формами, валидация. Практическое занятие 5. Работа над проектом «Разработка сайта средствами HTML и JS».	ПР, У
3	Раздел 3. Серверное web- программирование. PHP и MySQL.	Практическое занятие 6. Основы PHP. Условные операторы. Понятие функции. Практическое занятие 7. Циклы и массивы. Практическое занятие 8. Запросы HTTP, параметры URL и формы HTML. Практическое занятие 9. Стандартные средства PHP. Практическое занятие 10. Использование файлов в PHP. Практическое занятие 11. СУБД MySQL. Разработка базы	ПР, У

№	Наименование раздела	Наименование практических работ	Форма текущего контроля
1	2	3	4
		данных web-приложения.	
		Практическое занятие 12. Взаимодействие PHP и MySQL.	
		Практическое занятие 13. Авторизация пользователей. Сессии.	
		Практическое занятие 14. Использование Cookie. Почтовые рассылки в PHP.	
		Практическое занятие 15. Работа над проектом «Расширение функциональности сайта средствами PHP и MySQL».	

2.4.3. Лабораторные занятия

Не предусмотрены.

2.4.4. Примерная тематика курсовых работ

Не предусмотрены.

2.4.5. Содержание самостоятельной работы

Примерная тематика рефератов

1. Основные понятия Интернет: World Wide Web, Веб-серверы, Веб-страницы, Протоколы, Браузеры, Адресация в сети Интернет.
2. Глобальные компьютерные сети: основные понятия, принципы функционирования. Каталоги ресурсов. Поисковые системы.
3. Понятие технологии клиент-сервер, Web-сервер, сервер БД, почтовый сервер, файловый сервер.
4. Хостинг. Бесплатный хостинг. FTP. Размещение Интернет-ресурса на сервере провайдера. Регистрация Интернет-ресурса в каталогах и поисковых системах.
5. Классификация веб-сайтов.
6. Протокол HTTP. Структура запросов и ответов.
7. Протокол SMTP (соединение, посылка письма).
8. Протокол IMAP4 (соединение, выбор ящика, проверка флагов)
9. Протокол POP3 (соединение, получение списка писем, статистики, получение письма).
10. Протокол IMAP4 (чтение удаление, перемещение писем).
11. Протокол NNTP (соединение, получение списка групп, списка новостей, посылка новости).
12. Протокол FTP (соединение, подготовка второго соединения)
13. Протокол FTP (просмотр содержимого папки, получение файлов).
14. Протокол HTTP (соединение, метод GET).
15. Протокол HTTP (метод POST, HTTP 1.1).
16. Протокол ICMP (ping).
17. Общие понятия языка разметки гипертекста – HTML.
18. Структура HTML-документа.
19. Ввод и оформление текста в HTML.
20. Создание списков в HTML.
21. Создание ссылок в HTML.
22. Встраивание изображений на веб-страницу средствами HTML.
23. Добавление мультимедийных файлов на веб-страницу средствами HTML.

24. Вставка фоновых изображений и фоновой музыки на веб-страницу средствами HTML.
25. Создание таблиц средствами языка HTML.
26. Создание фреймов в HTML.
27. Конструирование форм в языке HTML.
28. Технология создания карты графических ссылок.
29. Основные правила Web-дизайна.
30. Понимание роли графики и анимации в Web-дизайне.
31. Общие подходы к дизайну сайта. Разработка макета страницы.
32. Введение в таблицы стилей и язык CSS.
33. Использование стиля при оформлении сайта. Возможности CSS.
34. Применение таблиц стилей CSS в форматировании текста.
35. Структурное форматирование в CSS.
36. Свойства для форматирования таблиц в CSS.
37. Пользовательские формы в CSS.
38. Позиционирование объектов в CSS.
39. Основы синтаксиса JavaScript и объектно-ориентированное программирование.
40. Внедрение JavaScript в код HTML страницы.
41. Выражения и операторы в JavaScript.
42. Создание объектов и функций в JavaScript.
43. Создание динамических веб-страниц средствами JavaScript.
44. Особенности языка PHP.
45. Переменные. Константы. Операторы в PHP. Циклы. Массивы. Работа со строками.
46. PHP. Операторы INCLUDE и REQUIRE. Особенности написания функций.
47. PHP. Работа с классами.
48. PHP. Регулярные выражения.
49. PHP. Работа с текстовыми файлами.
50. PHP. Функции управления сессиями.
51. PHP. Обработка входных данных.
52. PHP. Доступ к базам данных.
53. Функции в PHP. Встроенные функции.
54. Работа с датой и временем в PHP.
55. Подключение к базе данных из PHP файла. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу. Передача параметров в запрос.
56. Методы передачи параметров между страницами (GET, POST). Обработка действий пользователя при помощи форм.
57. Аутентификация пользователей средствами PHP.
58. Аутентификация пользователей средствами Web-сервера.
59. PHP. Способы управления сессиями. Работа с теневыми ссылками.
60. Создание HTML-страниц средствами PHP.
61. Связь PHP и HTML.
62. Работа с базой данных MySQL.
63. Архитектура базы данных MySQL (таблицы, связи, триггеры).
64. Принципы хранения информации в базах данных MySQL.
65. Важность безопасности в Веб-приложениях
66. Обзор технологий, используемых для обеспечения безопасности

2.4.6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

На самостоятельную работу студентов отводится 25 часов учебного времени в 7 семестре.

Самостоятельная работа студента в процессе освоения дисциплины включает:

- изучение основной и дополнительной литературы по курсу;
- самостоятельное изучение некоторых вопросов (конспектирование);
- работу с электронными учебными ресурсами;
- изучение Интернет-ресурсов;
- подготовку к практическим занятиям,
- самостоятельное выполнение домашних заданий,
- подготовку реферата по одной из тем курса.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Студент должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Студенты для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (лабораторным) занятиям. Желательно, чтобы конспекты лекций записывались в логической последовательности изучения курса и содержались в одной тетради.

Наименование раздела, темы	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
Раздел 1. HTML и CSS. Основы создания сайта.	<ol style="list-style-type: none">1. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен — СПб. : БХВ-Петербург, 2015. — 336 с. — ISBN 5-9775-0039-4.2. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с. : ил. ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=429078.3. Маркин, А.В. Основы web-программирования на PHP : учебное пособие / А.В. Маркин, С.С. Шкарин. - М. : Диалог-МИФИ, 2012. - 252 с. - ISBN 978-5-86404-241-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=229742.4. Баженова, И.Ю. SQL и процедурно-ориентированные языки / И.Ю. Баженова. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 167 с. : ил. - Библиогр. в кн. - ISBN 5-94774-539-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428934.5. Пархимович, М.Н. Основы интернет-технологий : учебное пособие / М.Н. Пархимович, А.А. Липницкий, В.А. Некрасова. - Архангельск : ИПЦ САФУ, 2013. - 366 с. : табл., ил. - Библиогр.: с. 351-352. - ISBN 978-5-261-00827-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=436379.
Раздел 2. Клиентское web-программирование. JavaScript.	<ol style="list-style-type: none">1. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен — СПб. : БХВ-Петербург, 2015. — 336 с. — ISBN 5-9775-0039-4.2. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с. : ил. ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=429078.3. Маркин, А.В. Основы web-программирования на PHP : учебное пособие / А.В. Маркин, С.С. Шкарин. - М. : Диалог-МИФИ, 2012. - 252 с. - ISBN 978-5-86404-241-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=229742.

Наименование раздела, темы	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
	<ol style="list-style-type: none"> <li data-bbox="480 232 1449 383">4. Баженова, И.Ю. SQL и процедурно-ориентированные языки / И.Ю. Баженова. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 167 с. : ил. - Библиогр. в кн. - ISBN 5-94774-539-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428934. <li data-bbox="480 383 1449 539">5. Пархимович, М.Н. Основы интернет-технологий : учебное пособие / М.Н. Пархимович, А.А. Липницкий, В.А. Некрасова. - Архангельск : ИПЦ САФУ, 2013. - 366 с. : табл., ил. - Библиогр.: с. 351-352. - ISBN 978-5-261-00827-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=436379.
Раздел 3. Серверное web-программирование. PHP и MySQL.	<ol style="list-style-type: none"> <li data-bbox="480 539 1449 600">1. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен — СПб. : БХВ-Петербург, 2015. — 336 с. — ISBN 5-9775-0039-4. <li data-bbox="480 600 1449 728">2. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с. : ил. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429078. <li data-bbox="480 728 1449 855">3. Маркин, А.В. Основы web-программирования на PHP : учебное пособие / А.В. Маркин, С.С. Шкарин. - М. : Диалог-МИФИ, 2012. - 252 с. - ISBN 978-5-86404-241-0 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=229742. <li data-bbox="480 855 1449 1005">4. Баженова, И.Ю. SQL и процедурно-ориентированные языки / И.Ю. Баженова. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 167 с. : ил. - Библиогр. в кн. - ISBN 5-94774-539-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428934. <li data-bbox="480 1005 1449 1153">5. Пархимович, М.Н. Основы интернет-технологий : учебное пособие / М.Н. Пархимович, А.А. Липницкий, В.А. Некрасова. - Архангельск : ИПЦ САФУ, 2013. - 366 с. : табл., ил. - Библиогр.: с. 351-352. - ISBN 978-5-261-00827-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=436379.

3. Образовательные технологии

Для улучшения усвоения учебного материала предусматривается использование активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся. В процессе преподавания применяются образовательные технологии развития критического мышления, проблемное обучение, проектное обучение, работа в малых группах.

3.1. Образовательные технологии при проведении лекций

№	Изучаемые темы дисциплины	Технологии, применяемые при проведении лекционных занятий	Кол-во часов
Раздел 1. HTML и CSS. Основы создания сайта.			
1	Основы HTML.	Аудиовизуальные технологии Технология проблемного обучения	1*
2	Каскадные таблицы стилей (CSS).	Аудиовизуальные технологии Технология проблемного обучения	1*
3	Использование таблиц для верстки.	Аудиовизуальные технологии Технология проблемного обучения	1*
4	Использование слоев в HTML. Верстка при помощи слоев.	Аудиовизуальные технологии Технология проблемного обучения	1*
Раздел 2. Клиентское web-программирование. JavaScript.			
5	Введение в JavaScript. Способы внедрения JS, стандартные функции. Способы внедрения JS, стандартные функции.	Аудиовизуальные технологии Технология проблемного обучения	1*
6	Типы переменных. Работа с переменными и операторами сравнения.	Аудиовизуальные технологии Технология проблемного обучения	1*
7	Управление web-объектами. Знакомство с событиями.	Аудиовизуальные технологии Технология проблемного обучения	1*
8	Работа с формами, валидация.	Аудиовизуальные технологии Технология проблемного обучения	1*
9	Классы и прототипы.	Аудиовизуальные технологии Технология проблемного обучения	1*
10	Cookies.	Аудиовизуальные технологии Технология проблемного обучения	1*
Раздел 3. Серверное web-программирование. PHP и MySQL.			
11	Основы PHP.	Аудиовизуальные технологии Технология проблемного обучения	1*
12	Условные операторы. Понятие функции.	Аудиовизуальные технологии Технология проблемного обучения	1*
13	Циклы и массивы.	Аудиовизуальные технологии Технология проблемного обучения	1*
14	Запросы HTTP, параметры URL и формы HTML.	Аудиовизуальные технологии Технология проблемного обучения	1*
15	Стандартные средства PHP.	Аудиовизуальные технологии Технология проблемного обучения	1*
16	Использование файлов в PHP.	Аудиовизуальные технологии Технология проблемного обучения	1*
17	СУБД MySQL. Разработка базы данных web-приложения.	Аудиовизуальные технологии Технология проблемного обучения	1*
18	Взаимодействие PHP и MySQL.	Аудиовизуальные технологии Технология проблемного обучения	1*
19	Авторизация пользователей. Сессии.	Аудиовизуальные технологии Технология проблемного обучения	1*
20	Использование Cookie. Почтовые рассылки в PHP.	Аудиовизуальные технологии Технология проблемного обучения	1*
Всего по дисциплине (в том числе интерактивное обучение*)			20 (20*)

3.2. Образовательные технологии при проведении практических занятий

	Изучаемые темы дисциплины	Технологии, применяемые при проведении практических занятий	Кол-во часов
Раздел 1. HTML и CSS. Основы создания сайта.			
1	Практическое занятие 1. Основы HTML. Каскадные таблицы стилей (CSS).	Технология проблемного обучения Технология работы в малых группах	2*
2	Практическое занятие 2. Использование таблиц для верстки. Верстка при помощи слоев.	Технология проблемного обучения Технология работы в малых группах	2*
Раздел 2. Клиентское web-программирование. JavaScript.			
3	Практическое занятие 3. Способы внедрения JS, стандартные функции. Работа с переменными и операторами сравнения.	Технология проблемного обучения Технология работы в малых группах	2*
4	Практическое занятие 4. Управление web-объектами. Знакомство с событиями. Работа с формами, валидация.	Технология проблемного обучения Технология работы в малых группах	2*
5	Практическое занятие 5. Работа над проектом «Разработка сайта средствами HTML и JS».	Технология проблемного обучения Технология работы в малых группах	2*
Раздел 3. Серверное web-программирование. PHP и MySQL.			
6	Практическое занятие 6. Основы PHP. Условные операторы. Понятие функции.	Технология проблемного обучения Технология работы в малых группах	2*
7	Практическое занятие 7. Циклы и массивы.	Технология проблемного обучения Технология работы в малых группах	2*
8	Практическое занятие 8. Запросы HTTP, параметры URL и формы HTML.	Технология проблемного обучения Технология работы в малых группах	2*
9	Практическое занятие 9. Стандартные средства PHP.	Технология проблемного обучения Технология работы в малых группах	2*
10	Практическое занятие 10. Использование файлов в PHP.	Технология проблемного обучения Технология работы в малых группах	2*
11	Практическое занятие 11. СУБД MySQL. Разработка базы данных web-приложения.	Технология проблемного обучения Технология работы в малых группах	2*
12	Практическое занятие 12. Взаимодействие PHP и MySQL.	Технология проблемного обучения Технология работы в малых группах	2*
13	Практическое занятие 13. Авторизация пользователей. Сессии.	Технология проблемного обучения Технология работы в малых группах	2*
14	Практическое занятие 14. Использование Cookie. Почтовые рассылки в PHP.	Технология проблемного обучения Технология работы в малых группах	2*
15	Практическое занятие 15. Работа над проектом «Расширение функциональности сайта средствами PHP и MySQL».	Технология проблемного обучения Технология работы в малых группах	2*
Всего по дисциплине (в том числе интерактивное обучение*)			30 (30*)

3.3. Образовательные технологии при проведении лабораторных занятий

Не предусмотрены.

4. Условия реализации программы учебной дисциплины

4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Реализация учебной дисциплины осуществляется в специально оборудованном кабинете лаборатории программного обеспечения компьютерных сетей, программирования и баз данных (г. Славянск-на-Кубани, ул. Зеленского, д. 10, М-24). Для самостоятельной подготовки по дисциплине служат библиотека и читальный зал.

Оборудование учебного кабинета:

Компьютеры — 30, компьютерный стол — 30, проектор, принтер, экран, локальная сеть, наглядные пособия, учебно-методические материалы, учебная мебель, доска учебная, выход в Интернет, система кондиционирования и вентиляции, система видеонаблюдения.

4.2. Перечень необходимого программного обеспечения

1. IDE для Web разработки NetBeans. — URL: <https://netbeans.org/downloads/>.
2. Java. — URL: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.
3. Notepad++. — URL: <https://notepad-plus-plus.org/>.
4. Atom. — URL: <https://atom.io/>.
5. Локальный Web сервер Denwer. — URL: <http://www.denwer.ru/>.
6. 7-zip (лицензия на англ. <http://www.7-zip.org/license.txt>).
7. Adobe Acrobat Reader (лицензия — <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>).
8. Adobe Flash Player (лицензия — <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>).
9. Apache OpenOffice (лицензия — <http://www.openoffice.org/license.html>).
10. FreeCommander (лицензия — <https://freecommander.com/ru/%d0%bb%d0%b8%d1%86%d0%b5%d0%bd%d0%b7%d0%b8%d1%8f/>).
11. Google Chrome (лицензия — https://www.google.ru/chrome/browser/privacy/eula_text.html).
12. LibreOffice (в свободном доступе).
13. Mozilla Firefox (лицензия — <https://www.mozilla.org/en-US/MPL/2.0/>).

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1 Основная литература

6. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен — СПб. : БХВ-Петербург, 2015. — 336 с. — ISBN 5-9775-0039-4.
7. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429078>.
8. Маркин, А.В. Основы web-программирования на PHP : учебное пособие / А.В. Маркин, С.С. Шкарин. - М. : Диалог-МИФИ, 2012. - 252 с. - ISBN 978-5-86404-241-0 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=229742>.
9. Баженова, И.Ю. SQL и процедурно-ориентированные языки / И.Ю. Баженова. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 167 с. : ил. - Библиогр. в кн. - ISBN 5-94774-539-9 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428934>.
10. Пархимович, М.Н. Основы интернет-технологий : учебное пособие / М.Н. Пархимович, А.А. Липницкий, В.А. Некрасова. - Архангельск : ИПЦ САФУ, 2013. - 366 с. : табл., ил. - Библиогр.: с. 351-352. - ISBN 978-5-261-00827-9 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=436379>.

5.2 Дополнительная литература

1. Гасанов, Э.В. Практикум по созданию Интернет-проектов. Основы языка программирования PHP / Э.В. Гасанов, С.Э. Гасанова. - М. : Издательство Книгодел, 2013. - Ч. 1. - 160 с. - (Свободное программное обеспечение). - ISBN 978-5-9659-0093-0 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=230535>.
2. Савельева, Н.В. Язык программирования PHP / Н.В. Савельева. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. — 330 с. : схем., ил. — Библиогр. в кн. ; То же [Электронный ресурс]. — URL: <http://biblioclub.ru/index.php?page=book&id=428975>.

5.3 Периодические издания

1. Вестник Московского Университета. Серия 15. Вычислительная математика и кибернетика. - URL: http://biblioclub.ru/index.php?page=journal_red&jid=237323
2. Инновации на основе информационных и коммуникационных технологий. - URL: <http://elibrary.ru/contents.asp?issueid=1438371>.
3. Информатика и образование. - URL: <http://dlib.eastview.com/browse/publication/18946/udb/1270>.
4. Информатика, вычислительная техника и инженерное образование. - URL: <http://elibrary.ru/contents.asp?issueid=1567393>.
5. Методические вопросы преподавания инфокоммуникаций в высшей школе. - URL: <http://elibrary.ru/contents.asp?titleid=55718>
6. Мир ПК. - URL: <http://dlib.eastview.com/browse/publication/64067/udb/2071>.

7. Открытые системы. СУБД. - URL:
<http://dlib.eastview.com/browse/publication/64072/udb/2071>
8. Программные продукты и системы. - URL: <http://dlib.eastview.com/browse/publication/64086/udb/2071>.
9. Computerworld Россия. - URL:
<http://dlib.eastview.com/browse/publication/64081/udb/2071>.
10. Windows IT Pro / Re. - URL: <http://dlib.eastview.com/browse/publication/64079/udb/2071>.

5.4 Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «Университетская библиотека ONLINE» : сайт. – URL: http://biblioclub.ru/index.php?page=main_ub_red.
2. ЭБС Издательства «Лань» : сайт. - URL: <http://e.lanbook.com>.
3. ЭБС «Юрайт» : [раздел «ВАША ПОДПИСКА: Филиал КубГУ (г. Славянск-на-Кубани)»] : сайт. - URL: <https://www.biblio-online.ru/catalog/E121B99F-E5ED-430E-A737-37D3A9E6DBFB>.
4. Научная электронная библиотека «eLibrary.ru» : сайт. – URL: <http://elibrary.ru/defaultx.asp>.
5. Базы данных компании «Ист Вью» [раздел: Периодические издания (на русском языке)] : сайт. – URL: <http://dlib.eastview.com>.
6. Федеральная информационная система «Единое окно доступа к образовательным ресурсам» : сайт. – URL: <http://windowedu.ru>.
7. Федеральный центр информационно-образовательных ресурсов (ФЦИОР) : сайт. - URL: <http://fcior.edu.ru> .
8. Единая коллекция цифровых образовательных ресурсов : сайт. – URL: <http://school-collection.edu.ru>.
9. Официальный интернет-портал правовой информации. Государственная система правовой информации : сайт. – URL: <http://publication.pravo.gov.ru/>.
10. Энциклопедиум : Энциклопедии. Словари. Справочники // ЭБС «Университетская библиотека ONLINE» : сайт. – URL: <http://enc.biblioclub.ru/>.

6. Методические указания обучающимся по освоению дисциплины

Дисциплина нацелена на формирование профессиональных компетенций (ПК 2.2, ПК 2.3, ПК 3.1).

Обучение студентов осуществляется по традиционной технологии (лекции, практики) с включением инновационных элементов.

Устное изложение учебного материала на лекции должно конспектироваться. Слушать лекцию нужно уметь – поддерживать своё внимание, понять и запомнить услышанное, уловить паузы. В процессе изложения преподавателем лекции студент должен выяснить все непонятные вопросы. Записывать содержание лекции нужно обязательно – записи помогают поддерживать внимание, способствуют пониманию и запоминанию услышанного, приводят знание в систему, служат опорой для перехода к более глубокому самостоятельному изучению предмета.

Методические рекомендации по конспектированию лекций:

- запись должна быть системной, представлять собой сокращённый вариант лекции преподавателя. Необходимо слушать, обдумывать и записывать одновременно;
- запись ведётся очень быстро, чётко, по возможности короткими выражениями;
- не прекращая слушать преподавателя, нужно записывать то, что необходимо усвоить. Нельзя записывать сразу же высказанную мысль преподавателя, следует её понять и после этого кратко записать своими словами или словами преподавателя. Важно, чтобы в ней не был потерян основной смысл сказанного;
- имена, даты, названия, выводы, определения записываются точно;
- следует обратить внимание на оформление записи лекции. Для каждого предмета заводится общая тетрадь. Отличным от остального цвета следует выделять отдельные мысли и заголовки, сокращать отдельные слова и предложения, использовать условные знаки, буквы латинского и греческого алфавитов, а также некоторые приёмы стенографического сокращения слов.

Практические занятия по дисциплине ОП.12 «Web-программирование» проводятся по схеме:

- устный опрос по теории в начале занятия;
- работа в группах по разрешению различных ситуаций по теме занятия;
- решение практических задач;
- индивидуальные задания для подготовки к практическим занятиям.

Цель практического занятия - научить студентов применять теоретические знания при решении практических задач на основе реальных данных.

На практических занятиях преобладают следующие методы:

- вербальные (преобладающим методом должно быть объяснение);
- практические (письменные задания, групповые задания и т. п.).

Важным для студента является умение рационально подбирать необходимую учебную литературу. Основными литературными источниками являются:

- библиотечные фонды филиала КубГУ;
- электронная библиотечная система «Университетская библиотека онлайн»;
- электронная библиотечная система издательства «Лань».

Поиск книг в библиотеке необходимо начинать с изучения предметного каталога и создания списка книг, пособий, методических материалов по теме изучения. Просмотр книги начинается с титульного листа, следующего после обложки. На нём обычно помещаются все основные данные, характеризующие книгу: название, автор, выходные данные, данные о переиздании и т.д. На обороте титульного листа даётся аннотация, в которой указывается тематика вопросов, освещённых в книге, определяется круг читателей, на который она рассчитана. Большое значение имеет предисловие книги, которое знакомит читателя с личностью автора, историей создания книги, раскрывает содержание. Прочив предисловие и получив общее представление о книге, следует обратиться к оглавлению. Оглавление книги знакомит обучаемого с содержанием и логической структурой книги, позволяет выбрать нужный материал для изучения. Год издания книги позволяет судить о новизне материала. Чем чаще книга издаётся, тем большую ценность она представляет. В книге могут быть примечания, которые содержат различные дополнительные сведения. Они печатаются вне основного текста и разъясняют отдельные вопросы. Предметные и алфавитные указатели значительно облегчают повторение изложенного в книге материала. В конце книги может располагаться вспомогательный материал. К нему обычно относятся инструкции, приложения, схемы, ситуационные задачи, вопросы для самоконтроля и т.д.

Для лучшего представления и запоминания материала целесообразно вести записи и конспекты различного содержания, а именно:

- пометки, замечания, выделение главного;
- план, тезисы, выписки, цитаты;
- конспект, рабочая запись, реферат, доклад, лекция и т.д.

Читать учебник необходимо вдумчиво, внимательно, не пропуская текста, стараясь понять каждую фразу, одновременно разбирая примеры, схемы, таблицы, рисунки, приведённые в учебнике. Одним из важнейших средств, способствующих закреплению знаний, является краткая запись прочитанного материала – составление конспекта. Конспект – это краткое связное изложение содержания темы, учебника или его части, без подробностей и второстепенных деталей. По своей структуре и последовательности конспект должен соответствовать плану учебника. Поэтому важно сначала составить план, а потом писать конспект в виде ответа на вопросы плана. Если учебник разделён на небольшие озаглавленные части, то заголовки можно рассматривать как пункты плана, а из текста каждой части следует записать те мысли, которые раскрывают смысл заголовка. Требования к конспекту:

- краткость, сжатость, целесообразность каждого записываемого слова;
- содержательность записи- записываемые мысли следует формулировать кратко, но без ущерба для смысла. Объём конспекта, как правило, меньше изучаемого текста в 7–15 раз;
- конспект может быть как простым, так и сложным по структуре – это зависит от содержания книги и цели её изучения.

Методические рекомендации по конспектированию:

- прежде чем начать составлять конспект, нужно ознакомиться с книгой, прочитать её сначала до конца, понять прочитанное;
- на обложке тетради записываются название конспектируемой книги и имя автора, составляется план конспектируемого текста;

- записи лучше делать при прочтении не одного-двух абзацев, а целого параграфа или главы;
- конспектирование ведётся не с целью иметь определённые записи, а для более полного овладения содержанием изучаемого текста, поэтому в записях отмечается и выделяется всё то новое, интересное и нужное, что особенно привлекло внимание;
- после того, как сделана запись содержания параграфа, главы, следует перечитать её, затем снова обратиться к тексту и проверить себя, правильно ли изложено содержание.

Техника конспектирования:

- конспектируя книгу большого объёма, запись следует вести в общей тетради;
- на каждой странице слева оставляют поля шириной 25–30 мм для записи коротких подзаголовков, кратких замечаний, вопросов;
- каждая страница тетради нумеруется;
- для повышения читаемости записи оставляют интервалы между строками, абзацами, новую мысль начинают с «красной» строки;
- при конспектировании широко используют различные сокращения и условные знаки, но не в ущерб смыслу записанного. Рекомендуется применять общеупотребительные сокращения, например: м.б. – может быть; гос. – государственный; д.б. – должно быть и т.д.
- не следует сокращать имена и названия, кроме очень часто повторяющихся;
- в конспекте не должно быть механического переписывания текста без продумывания его содержания и смыслового анализа.

Самостоятельная работа студентов является важнейшей формой учебно-познавательного процесса. Основная цель самостоятельной работы студента при изучении дисциплины – закрепить теоретические знания, полученные в ход лекционных занятий, а также сформировать практические навыки подготовки в области математики.

Самостоятельная работа студента в процессе освоения дисциплины ОП.12 «Web-программирование» включает:

- изучение основной и дополнительной литературы по курсу;
- работу с электронными учебными ресурсами;
- изучение материалов периодической печати, Интернет-ресурсов;
- подготовку к контрольной работе;
- индивидуальные и групповые консультации по наиболее сложным вопросам.

На самостоятельную работу студентов отводится 20 часов в 6 семестре и 20 часов в 7 семестре учебного времени. Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Студент должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями. Студенты для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (семинарским) занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Ведение конспекта является необходимым видом учебной деятельности. Поскольку конспект:

- в полном объеме оценивается как разновидность письменного ответа на изучаемые вопросы;
- служит базой для устного ответа на семинаре по одному из вопросов рассматриваемого плана;
- сведения из конспекта могут выступать в качестве источника дополнений к ответам других студентов.

Организация текущего контроля знаний, умений и навыков обучающихся осуществляется путём устного опроса, контрольных работ, подготовки рефератов.

Реферат — продукт самостоятельной работы студента, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее. Реферат оценивается по количеству обработанных источников, глубине анализа проблемы, качестве обоснования авторской позиции, глубине раскрытия темы.

Для написания реферата необходимо выбрать тему, согласовать ее с преподавателем, подобрать несколько источников по теме, выполнить анализ источников по решению проблемы, обосновать свою точку зрения на решение проблемы.

Формой итогового контроля является дифференцированный зачет.

7. Оценочные средства для контроля успеваемости

7.1. Паспорт фонда оценочных средств

№	Изучаемые темы дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
Семестр 6			
Раздел 1. HTML и CSS. Основы создания сайта.			
1	Основы HTML.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
2	Каскадные таблицы стилей (CSS).	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
3	Использование таблиц для верстки.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
4	Работа с дизайнерским макетом сайта.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
5	Использование слоев в HTML.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
6	Верстка при помощи слоев.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
Раздел 2. Клиентское web-программирование. JavaScript.			
7	Введение в JavaScript.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
8	Способы внедрения JS, стандартные функции.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
9	Работа с переменными и операторами сравнения.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
10	Типы переменных.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
11	Управление web-объектами. Знакомство с событиями.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
12	Работа с формами, валидация.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
13	Классы и прототипы.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
14	Cookies.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
Семестр 7			
Раздел 3. Серверное web-программирование. PHP и MySQL.			
1	Основы PHP.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
2	Условные операторы. Понятие функции.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
3	Циклы и массивы.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
4	Запросы HTTP, параметры URL и формы HTML.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
5	Стандартные средства PHP.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
6	Использование файлов в PHP.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
7	СУБД MySQL. Разработка базы данных web-приложения.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
8	Взаимодействие PHP и MySQL.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
9	Авторизация пользователей. Сессии.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа
10	Использование Cookie. Почтовые рассылки в PHP.	ПК 2.2, ПК 2.3, ПК 3.1	Реферат, практическая работа

7.2. Критерии оценки результатов обучения

Результаты обучения оцениваются по традиционной шкале:

- **«отлично»** выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;
- **«хорошо»** выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- **«удовлетворительно»** выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки

базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

- **«неудовлетворительно»** выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

7.3. Оценочные средства для проведения текущей аттестации

Текущий контроль может проводиться в форме:

- фронтальный опрос;
- индивидуальный устный опрос;
- письменный контроль;
- защита реферата.

Форма аттестации	Знания	Умения	Владения (навыки)	Личные качества студента	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по теме (определения, правила, основные положения)	Оценка умения различать конкретные понятия, формулировать и применять определения и правила	Оценка навыков работы с конспектами лекций, с учебниками, защита лабораторных работ	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Рефераты	Контроль знаний по теме (отдельные вопросы материала)	Оценка умения анализировать, обобщать и систематизировать материал по теме	Оценка навыков работы с литературными источниками	Оценка способности к самостоятельной работе и анализу литературных источников	Темы рефератов прилагаются

Примерные вопросы для устного опроса

Раздел 1. HTML и CSS. Основы создания сайта.

1. Структура HTML страницы: заголовок и тело документа.
2. Использование в html комментариев и спецсимволов.
3. Теги для оформления текста.
4. Работа со стилем текста, его разбиение на блоки.
5. Гиперссылки, их объявление на странице.
6. Иллюстрации на веб-странице и их правильное использование.
7. Обмен с сервером. Формы и элементы управления.
8. Понятие и назначение CSS.
9. Синтаксис CSS.
10. CSS: селекторы (id, class, tag).
11. CSS: создание стилей и задание их свойств.
12. CSS: вложенность, наследование и группы свойств.
13. CSS: приоритеты применения стилей.
14. CSS: псевдоклассы и псевдоэлементы.

15. Теги HTML для создания таблиц.
16. Свойства таблицы и их использование для верстки сайтов.
17. Задание шаблона сайта при помощи таблиц.
18. Объединение ячеек в таблицах в HTML.
19. Вложенные таблицы в HTML.
20. Использование атрибутов отображения таблиц в HTML.
21. Табличная верстка в HTML.
22. Графический редактор GIMP.
23. Выделение частей макета. Нарезка макета.
24. Табличная верстка дизайнерского макета.
25. Правила добавления блоков макета при верстке.
26. Слои и их использование. Верстка слоями в HTML.
27. Методика верстки. Теги (div и span) в HTML.
28. Использование атрибутов тегов. Отступы элементов (margin и padding) в HTML.
29. Размещение блоков на странице в HTML.
30. Применение overflow и clear в реальной вёрстке.
31. Заполнение разметки частями нарезанного макета.
32. Методы размещения элементов при помощи слоев.

Раздел 2. Клиентское web-программирование. JavaScript.

1. История JavaScript.
2. Понятие JavaScript.
3. Область применения JavaScript.
4. Литералы, переменные, выражения в JavaScript.
5. JavaScript-код на странице.
6. Подключение внешнего JavaScript файла.
7. Знакомство с alert() и console() в JavaScript.
8. Самые распространенные ошибки в JavaScript.
9. Простые математические операции в JavaScript.
10. Использование оператора if & else в JavaScript.
11. Оператор switch в JavaScript..
12. Обработка ошибок Try & catch.
13. Знакомство с JavaScript-методами: Math, String, Window.
14. Числа в JavaScript.
15. Строки в JavaScript.
16. Массивы в JavaScript.
17. Регулярные выражения в JavaScript.
18. Управление web-объектами в JavaScript.
19. Событие CLICK в JavaScript.
20. Событие MouseOver в JavaScript.
21. Работа с окнами в JavaScript.
22. Получение параметра ввода пользователя в JavaScript.
23. Изменения параметра ввода, преобразование в JavaScript.
24. Конструкторы в JavaScript.
25. Прототип и наследование свойств в JavaScript.
26. Создание класса, обращение к методам класса, удаление класса в JavaScript.

27. История cookies.
28. Безопасность и ограничения применения cookie.
29. Область применения cookie.
30. Чтение cookie. Удаление cookie.

Раздел 3. Серверное web-программирование. PHP и MySQL.

1. Назначение PHP. Особенности языка.
2. Установка и настройка связки Apache, PHP, MySQL. Сборка Denwer.
3. Синтаксис языка PHP.
4. Переменные в языке PHP. Типы данных.
5. Преобразование типов данных.
6. Константы.
7. Арифметические операции. Интерпретация арифметических операций для строк.
8. Логические операции. Интерпретация логических операций для строк.
9. Условия при выполнении программы. Ветвление алгоритма.
10. Оператор if. Оператор switch.
11. Функции в языке PHP. Использование функций. Объявление и определение функции.
12. Передаваемые в функцию параметры. Задание значения по умолчанию для передаваемых параметров.
13. Возвращение значения из функции. Оператор return.
14. Рекурсивный вызов функции. Применение рекурсии.
15. Область видимости и время жизни переменных в программе на PHP.
16. Использование циклов в PHP.
17. Циклы while и do..while.
18. Циклы for и foreach.
19. Организация бесконечного цикла.
20. Оператор выхода из цикла. Оператор прерывания итерации цикла.
21. Массивы. Объявление массивов в языке PHP.
22. Использование циклов для работы с массивом.
23. Функции для работы с массивами.
24. Многомерные массивы в PHP.
25. Предопределенные массивы.
26. Принципы работ протокола HTTP. Запросы GET и POST.
27. Метод GET. Передача пользовательских параметров через URL. Параметры запроса в URL.
28. Обработка параметров, полученных методом GET в PHP.
29. Отправка параметров методом POST. Обработка отправки запросов через HTML форму.
30. Преимущества и недостатки методов GET и POST.
31. Документация по PHP в сети Internet.
32. Встроенные функции языка PHP.
33. Вычисление математических функций.
34. Обработка строк. Регулярные выражения.
35. Функции для работы с массивами.

36. Работа с переменными даты и времени.
37. Встроенные константы.
38. Переменные среды.
39. Принципы работы с файлами в PHP.
40. Режимы открытия файла. Чтение и запись.
41. Средства языка для чтения и записи файлов.
42. Обработка ошибок при работе с файлами.
43. Организация загрузки файла на сайт.
44. Работы с каталогами, расположенными на сервере.
45. Получение списка файлов и директорий.
46. Использование СУБД MySQL для веб-разработки.
47. Установка и конфигурация MySQL под ОС Windows.
48. Программы для работы с сервером баз данных MySQL.
49. Установка PhpMyAdmin. Использование PhpMyAdmin для разработки базы данных.
50. Экспорт и импорт баз данных.
51. Проектирование базы данных. Первичные и вторичные ключи.
52. Графическое представление базы данных. ER-диаграмма.
53. Расширения языка PHP для MySQL.
54. Соединение с сервером баз данных.
55. Запросы для получения информации.
56. Обработка результата запроса.
57. Обеспечение безопасности данных.
58. Использование сессий.
59. Создание сессии.
60. Хранение паролей в БД. Хеширование и алгоритм md5.
61. Чтение данных сессии. Закрытие сессии.
62. Хранение пользовательских данных в Cookie.
63. Типы cookie: постоянные, временные.
64. Работы с cookie: создание, чтение, удаление.
65. Расширения языка PHP для работы с электронной почтой.
66. Создание электронного письма средствами PHP.

Примерные темы рефератов

1. Основные понятия Интернет: World Wide Web, Веб-серверы, Веб-страницы, Протоколы, Браузеры, Адресация в сети Интернет.
2. Глобальные компьютерные сети: основные понятия, принципы функционирования. Каталоги ресурсов. Поисковые системы.
3. Понятие технологии клиент-сервер, Web-сервер, сервер БД, почтовый сервер, файловый сервер.
4. Хостинг. Бесплатный хостинг. FTP. Размещение Интернет-ресурса на сервере провайдера. Регистрация Интернет-ресурса в каталогах и поисковых системах.
5. Классификация веб-сайтов.
6. Протокол HTTP. Структура запросов и ответов.
7. Протокол SMTP (соединение, посылка письма).

8. Протокол IMAP4 (соединение, выбор ящика, проверка флагов)
9. Протокол POP3 (соединение, получение списка писем, статистики, получение письма).
10. Протокол IMAP4 (чтение удаление, перемещение писем).
11. Протокол NNTP (соединение, получение списка групп, списка новостей, посылка новости).
12. Протокол FTP (соединение, подготовка второго соединения)
13. Протокол FTP (просмотр содержимого папки, получение файлов).
14. Протокол HTTP (соединение, метод GET).
15. Протокол HTTP (метод POST, HTTP 1.1).
16. Протокол ICMP (ping).
17. Общие понятия языка разметки гипертекста – HTML.
18. Структура HTML-документа.
19. Ввод и оформление текста в HTML.
20. Создание списков в HTML.
21. Создание ссылок в HTML.
22. Встраивание изображений на веб-страницу средствами HTML.
23. Добавление мультимедийных файлов на веб-страницу средствами HTML.
24. Вставка фоновых изображений и фоновой музыки на веб-страницу средствами HTML.
25. Создание таблиц средствами языка HTML.
26. Создание фреймов в HTML.
27. Конструирование форм в языке HTML.
28. Технология создания карты графических ссылок.
29. Основные правила Web-дизайна.
30. Понимание роли графики и анимации в Web-дизайне.
31. Общие подходы к дизайну сайта. Разработка макета страницы.
32. Введение в таблицы стилей и язык CSS.
33. Использование стиля при оформлении сайта. Возможности CSS.
34. Применение таблиц стилей CSS в форматировании текста.
35. Структурное форматирование в CSS.
36. Свойства для форматирования таблиц в CSS.
37. Пользовательские формы в CSS.
38. Позиционирование объектов в CSS.
39. Основы синтаксиса JavaScript и объектно-ориентированное программирование.
40. Внедрение JavaScript в код HTML страницы.
41. Выражения и операторы в JavaScript.
42. Создание объектов и функций в JavaScript.
43. Создание динамических веб-страниц средствами JavaScript.
44. Особенности языка PHP.
45. Переменные. Константы. Операторы в PHP. Циклы. Массивы. Работа со строками.
46. PHP. Операторы INCLUDE и REQUIRE. Особенности написания функций.
47. PHP. Работа с классами.
48. PHP. Регулярные выражения.
49. PHP. Работа с текстовыми файлами.
50. PHP. Функции управления сеансами.
51. PHP. Обработка входных данных.

52. PHP. Доступ к базам данных.
53. Функции в PHP. Встроенные функции.
54. Работа с датой и временем в PHP.
55. Подключение к базе данных из PHP файла. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу. Передача параметров в запрос.
56. Методы передачи параметров между страницами (GET, POST). Обработка действий пользователя при помощи форм.
57. Аутентификация пользователей средствами PHP.
58. Аутентификация пользователей средствами Web-сервера.
59. PHP. Способы управления сессиями. Работа с теневыми посылками.
60. Создание HTML-страниц средствами PHP.
61. Связь PHP и HTML.
62. Работа с базой данных MySQL.
63. Архитектура базы данных MySQL (таблицы, связи, триггеры).
64. Принципы хранения информации в базах данных MySQL.
65. Важность безопасности в Веб-приложениях
66. Обзор технологий, используемых для обеспечения безопасности

7.4. Оценочные средства для проведения промежуточной аттестации

Форма аттестации	Знания	Умения	Владение (навыки)	Личные качества студента	Примеры оценочных средств
Зачет, Экзамен	Контроль знания базовых положений	Оценка умения понимать специальную терминологию	Оценка навыков логического сопоставления и характеристики объектов	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения решать типовые задачи в области компьютерных сетей	Оценка навыков при решении задач в области компьютерных сетей	Оценка способности грамотно и четко излагать ход решения задач и интерпретировать результаты	Задачи прилагаются

7.4.1. Примерные вопросы для проведения промежуточной аттестации

Промежуточная аттестация в каждом семестре проводится в форме защиты итогового проекта, выполняемого на практических занятиях и в ходе самостоятельной работы. В процессе защиты преподаватель оценивает функциональность и качество разработанного web-ресурса, степень самостоятельности выполнения студентом проекта, уровень владения использованными технологиями. В 6 семестре при выполнении проекта используются возможности HTML и JavaScript, в седьмом, — технологии программирования на стороне сервера средствами PHP и MySQL. В процессе защиты преподаватель может задать студенту 2–3 вопроса из перечня для текущего контроля. Проекты предоставляются в электронном виде.

7.4.2. Примерные задачи для проведения промежуточной аттестации

Задание 1. Разработать сайт средствами HTML и JavaScript по материалам курсовой работы (выпускной квалификационной работы). Выполнить размещение сайта на локальном web-сервере.

Задание 2. Расширить функциональные возможности сайта, разработанного в 6 семестре, средствами PHP и MySQL. Выполнить размещение сайта на локальном web-сервере, предварительно сконфигурированном для работы с PHP и MySQL.

8. Дополнительное обеспечение дисциплины

Дополнительный материал, расширяющий и углубляющий лекции преподавателя, можно найти в источниках, перечисленных в разделе «Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины» (см. раздел 5).

Также, по настоятельной просьбе УМО филиала, ниже приводится фрагмент лекционного занятия.

Каскадные таблицы стилей

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — технология описания внешнего вида документа, оформленного языком разметки.

Преимущественно используется как средство оформления веб-страниц в формате HTML и XHTML, но может применяться с любыми видами документов в формате XML, включая SVG и XUL.

Каскадные таблицы стилей используются создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления веб-документа. Основной целью разработки CSS являлось разделение содержимого (написанного на HTML или другом языке разметки) и оформления документа (написанного на CSS). Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Что такое CSS?

Каскадные таблицы стилей (Cascading Style Sheets, CSS) — это стандарт, определяющий представление данных в браузере. Если HTML предоставляет информацию о структуре документа, то таблицы стилей сообщают как он должен выглядеть.

Стиль — это совокупность правил, применяемых к элементу гипертекста и определяющих способ его отображения. Стиль включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице.

Таблица стилей — это совокупность стилей, применимых к гипертекстовому документу.

Каскадирование — это порядок применения различных стилей к веб-странице. Браузер, поддерживающий таблицы стилей, будет последовательно применять их в

соответствии с приоритетом: сначала связанные, затем внедренные и, наконец, встроенные стили. Другой аспект каскадирования — *наследование (inheritance)*, — означает, что если не указано иное, то конкретный стиль будет применен ко всем дочерним элементам гипертекстового документа. Например, если вы примените определенный цвет текста в теге `<div>`, то все теги внутри этого блока будут отображаться этим же цветом.

Использование каскадных таблиц дает возможность разделить содержимое и его представление и гибко управлять отображением гипертекстовых документов путем изменения стилей.

Официальная информация о спецификации Cascading Style Sheets всегда доступна по адресу <http://www.w3.org/Style/CSS/>

Общий синтаксис таблиц стилей

Таблицы стилей строятся в соответствии с определенным порядком (синтаксисом), в противном случае они не могут нормально работать. Таблицы стилей состояются из определенных частей (рис. 1):

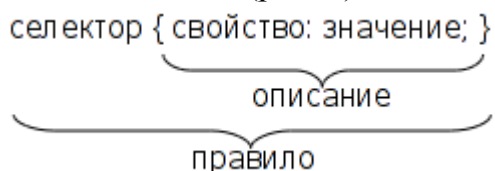


Рис. 1. Синтаксис описания стиля CSS

- *Селектор (Selector)*. Селектор — это элемент, к которому будут применяться назначаемые стили. Это может быть тег, класс или идентификатор объекта гипертекстового документа.
- *Свойство (Property)*. Свойство определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д.
- *Значение (Value)*. Значения — это фактические числовые или строковые константы, определяющие свойство селектора.
- *Описание (Declaration)*. Совокупность свойств и их значений.
- *Правило (Rule)*. Полное описание стиля (селектор + описание).

Таким образом, таблица стилей — это набор правил, задающих значения свойств селекторов, перечисленных в этой таблице. Общий синтаксис описания правила выглядит так:

```
селектор[, селектор[, ...]] {свойство: значение;}
```

Регистр символов значения не имеет, порядок перечисления селекторов в таблице и свойств в определении не регламентирован.

Правила CSS

Итак, каскадная таблица стилей — это набор правил форматирования тегов HTML. Приведем несколько примеров написания таких правил:

1. Основной текст с выравниванием по ширине, абзацный отступ 30px, гарнитура (шрифт) — Serif, кегль (размер шрифта) — 14px:

```
p {  
  text-align: justify;  
  text-indent: 30px;
```

```
font-family: Serif;
font-size: 14px;
}
```

Это правило будет применено ко всем тегам `<p>`.

2. Синий цвет для заголовков с первого по третий уровень:

```
h1, h2, h3 {
  color: blue; /* тоже самое, что и #0000FF */
}
```

3. Таблицы и изображения выводить без обрамления:

```
table, img {border: none;}
```

4. Ссылки в элементах списков показывать без подчеркивания:

```
li a {text-decoration: none;}
```

5. Внутренние отступы слева и справа для блоков (`<div>`), заголовков таблиц и ячеек таблиц установить в 10px и залить фон желтым цветом:

```
div, th, td {
  padding-left: 10px;
  padding-right: 10px;
  background-color: yellow;
}
```

6. Все ссылки в документе отображать черным цветом и полужирным шрифтом, а в основном тексте и списках — обычным, а также выделять их зеленым цветом и подчеркивать только при наведении курсора (в описании правил использован псевдоэлемент `a:hover`).

```
a {color: black; font-weight: bold;}
p a, li a {font-weight: normal; text-decoration: none;}
p a:hover, li a:hover {
  color: #00FF00; text-decoration: underline;
}
```

Классы

Стандарт CSS представляет возможности создания именованных стилей — стилевых классов. Это позволяет ответить на такой, например, вопрос: Как применить разные стили к одному и тому же селектору?

Предположим, что в документе вам нужны два различных вида основного текста — один без отступа, второй — с левым отступом и шрифтом красного цвета. Для этого нужно создать правила для каждого из них, например так:

```
p {margin-left: 0;}
p.warn {margin-left: 40px; color: #FF00;}
```

Для применения созданного класса его имя нужно указать в атрибуте `class` для выбранных абзацев:

```
<p class="warn">Красный текст с отступом слева</p>
```

Общий синтаксис описания класса:

```
селектор.имя_класса {описание}
```

При создании класса селектор можно не указывать, тогда это правило можно применять к любому селектору, поддерживающему тот же набор свойств.

Вот несколько примеров:

Правило:

```
.solid_blue {color: blue;}
```

Использование:

```
<p class="solid_blue">Синий текст абзаца</p>
```

```
<li class="solid_blue">Синий текст элемента списка</li>
```

Правило:

```
h1.bigsans {font-family: Sans; font-size: 1.5em;}
```

```
h1.smallserif {font-family: Serif; font-size: .84em;}
```

Использование:

```
<h1 class="bigsans">Большой, но рубленый</h1>
```

```
<h1 class="smallserif">Маленький, но с засечками</h1>
```

Идентификаторы

В качестве селектора может выступать идентификатор элемента гипертекста, указанный в атрибуте id. Для назначения стилей таким элементам используется синтаксис, аналогичный описанию классов, но вместо точки ставится знак # (“решетка”). Например:

```
div#content {  
    position: absolute;  
    top: 10px;  
    left: 10%;  
    right: 10%;  
    border: solid 1px silver;  
}  
...
```

```
<div id="content">Текст</div>
```

Следует помнить, что идентификаторы элементов должны быть уникальны в пределах документа.

Группировка свойств

Группировка (grouping) состоит в объединении значений родственных свойств. При этом таблица стилей становится более компактной, но предъявляются более жесткие требования к описанию правил. Ниже приведен пример обычного стиля, задающего отступы:

```
div {  
    margin-left: 10px;  
    margin-top: 5px;  
    margin-right: 40px;  
    margin-bottom: 15px;  
}
```

Это же правило можно переписать с группировкой в следующем виде:

```
div {margin: 5px 40px 15px 10px;} /*порядок: top right bottom left*/
```

Оба стиля будут отображаться одинаково.

Группировка может применяться для таких свойств, как padding, font, border, background и еще некоторых (см. документацию CSS)

Использование в веб-страницах

Существует три способа применения таблицы стилей к документу HTML:

- *Встраивание (Inline)*. Этот метод позволяет применить стиль к заданному тегу HTML.
- *Внедрение (Embedded)*. Внедрение позволяет управлять стилями страницы целиком.
- *Связывание (Linked или External)*. Связанная таблица стилей позволяет вынести описание стилей во внешний файл, ссылаясь на который можно контролировать отображение всех страниц сайта.

Встроенные стили

Встраивание стилей предоставляет максимальный контроль над всеми элементами веб-страницы. Встроенный стиль применяется к любому тегу HTML с помощью атрибута *style* следующим образом:

```
<p style="font: 12pt Courier">Это текст с кеглем 12 точек и гарнитурой Courier</P>
```

Пример:

```
<div style="font-family: Garamond; font-size: 18 pt;">
```

Весь текст в этом разделе имеет размер 18 точек и шрифт Garamond.

```
<span style="color:#ff3300;">
```

А этот фрагмент еще и выделен красным цветом.

```
</div>
```

Встроенные стили полезны, когда необходима тонкая настройка отображения некоторого элемента страницы или небольшой веб-страницы.

Внедренные стили

Внедренные стили используют тег `<style>`, который обычно размещают в заголовке HTML-документа (`<head>...</head>`):

```
<html>
```

```
<head>
```

```
...
```

```
<style>
```

```
    правила CSS
```

```
</style>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

Связанные таблицы стилей

Связанные (linked), или *внешние (external)* таблицы стилей — наиболее удобное решение, когда речь идет об оформлении целого сайта. Описание правил помещается в отдельный файл (обычно, но не обязательно, с расширением *.css*). С помощью тега `<link>` выполняется связывание этой таблицы стилей с каждой страницей, где ее необходимо применить, например так:

```
<link rel=stylesheet href="sample.css" type="text/css">
```

Любая страница, содержащая такую связь, будет оформлена в соответствии со стилями, указанными в файле *sample.css*. Следует отметить, что файл со стилями физически может находиться на другом веб-сервере, тогда в *href* нужно указать абсолютный путь к нему.

Проблемы с браузерами

Обязательно просматривайте страницы с таблицами стилей в различных браузерах. Это связано с тем, что разные браузеры могут по-разному интерпретировать одно и то же правило, а некоторые свойства и/или значения и вовсе не поддерживать. Следует также тестировать страницы с отключенными стилями (например, в текстовых браузерах), чтобы убедиться, что страница читабельна.

И снова каскадирование

Если вам нужна сотня-другая-третья страниц HTML — используйте внешнюю, глобальную, таблицу стилей. Если некоторые из этих страниц требуют корректировки общего оформления — используйте внедренный стиль. А если на странице нужно явно изменить оформление одного-двух элементов, то применяйте встроенные стили. Именно в таком порядке происходит перекрытие стилей при каскадировании, схематично это можно представить так: *связанные стили -> внедренные стили -> встроенные стили*

Аппаратно-зависимые стили

Таблицы стилей могут применяться для управления отображением содержимого в зависимости от используемого устройства вывода (монитор, проектор, устройство печати, звуковой синтезатор и т.п.). Для этого в описание стилей включить тип устройства, например так:

```
@media print { /* печатающее устройство */  
  BODY { font-size: 10pt; }  
}
```

```
@media screen { /* монитор */  
  BODY { font-size: 12pt; }  
}
```

```
@media screen, print {  
  BODY { line-height: 1.2; }  
}
```

```
@media all {  
  BODY { margin: 1pt; }  
}
```

Как видно из примера, вся таблица разбивается на секции, каждая из которых начинается со слова `@media`, за которым следует название класса устройств и далее, в фигурных скобках, непосредственно описание стилей.

Можно разделить таблицы стилей иначе, указав тип устройства в теге `<link>`:

```
<link rel=stylesheet href="sample.css" type="text/css" media="screen">
```

Свойства CSS

В табл.1 перечислены некоторые часто используемые свойства CSS и их назначение.

Таблица 1. Свойства элементов CSS

Имя	Значения	Описание
background	[background-color background-image background-repeat background-attachment background-position] inherit	Управление фоном элемента
background-color	<color> transparent inherit	Цвет фона
background-image	<uri> none inherit	Фоновое изображение

Имя	Значения	Описание
background-position	[[<percentage> <length>]{1,2} [[top center bottom] [left center right]]] inherit	Положение фоновой картинки
background-repeat	repeat repeat-x repeat-y no-repeat inherit	Повторение фоновой картинки
border	[border-width border-style <color>] inherit	Границы элемента
border-collapse	collapse separate inherit	Объединение/разделение смежных границ
border-color	<color>{1,4} transparent inherit	Цвет границы
border-style	<border-style>{1,4} inherit	Стиль линии границы
border-top border-right border-bottom border-left	[border-top-width border-style <color>] inherit	Управление стилем заданной границы
border-width	<border-width>{1,4} inherit	Толщина линии границы
bottom	<length> <percentage> auto inherit	Низ элемента
clear	none left right both inherit	Запрет заполнения свободного пространства рядом с элементом
clip	<shape> auto inherit	Обрезка содержимого элемента
color	<color> inherit	Цвет содержимого
cursor	[[<uri> ,]* [auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help]] inherit	Форма курсора
display	inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption none inherit	Способ отображения элемента
empty-cells	show hide inherit	Отображение пустых ячеек таблицы
float	left right none inherit	Свободное размещение элемента
font	[[font-style font-variant font-weight]? font-size [/ line-height]? font-family] caption icon menu message-box small-caption status-bar inherit	Управление шрифтом
font-family	[[<family-name> <generic-family>],]* [<family-name> <generic-family>] inherit	Гарнитура
font-size	<absolute-size> <relative-size> <length> <percentage> inherit	Кегль
font-style	normal italic oblique inherit	Стиль шрифта
font-variant	normal small-caps inherit	Варианты отображения шрифта
font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	Толщина шрифта
height	<length> <percentage> auto inherit	Ширина элемента
left	<length> <percentage> auto inherit	Положение левой границы элемента
line-height	normal <number> <length> <percentage> inherit	Высота строки
list-style	[list-style-type list-style-position list-style-image] inherit	Стиль списка
margin	<margin-width>{1,4} inherit	Внешний отступ
margin-top margin-right margin-bottom margin-left	<margin-width> inherit	Внешний отступ по заданной стороне

Имя	Значения	Описание
padding	<padding-width> { 1,4 } inherit	Внутренний отступ
padding-top padding-right padding-bottom padding-left	<padding-width> inherit	Внутренний отступ по заданной стороне
position	static relative absolute fixed inherit	Позиционирование элемента
right	<length> <percentage> auto inherit	Положение правой границы
text-align	left right center justify <string> inherit	Выравнивание текстового блока
text-decoration	none [underline overline line-through blink] inherit	Текстовые эффекты
text-indent	<length> <percentage> inherit	Абзацный отступ
text-transform	capitalize uppercase lowercase none inherit	Начертание текста
top	<length> <percentage> auto inherit	Положение верхней границы элемента
vertical-align	baseline sub super top text-top middle bottom text-bottom <percentage> <length> inherit	Вертикальное выравнивание в пределах блока
visibility	visible hidden collapse inherit	Управление видимостью элемента
white-space	normal pre nowrap inherit	Управление пробелами между словами
width	<length> <percentage> auto inherit	Ширина элемента
z-index	auto <integer> inherit	Порядок перехода по клавише Tab

Позиционирование элементов

Влияние на внешний вид оказывает свойство `position`. Это свойство в сочетании со свойствами `left`, `top`, `right`, `bottom`, `display`, `clear` и ряда других позволяет управлять положением элементов на странице и порядком их вывода. Свойство `position` может принимать такие значения:

`static` — нормальное положение

Данный блок является обычным блоком, он отображается согласно общим правилам. Свойства `'left'` и `'top'` не применяются.

`relative` — относительное позиционирование

Положение блока рассчитывается в соответствии с нормальным потоком вывода. Затем блок смещается относительно своего нормального (`static`) положения.

`absolute` — абсолютное позиционирование

Положение блока (возможно и размер) указывается с помощью свойств `'left'`, `'right'`, `'top'` и `'bottom'`. Они указывают величину смещения относительно контейнера блока. Абсолютно позиционируемые блоки изымаются из нормального потока. Это значит, что они не влияют на размещение последующих элементов того же уровня.

`fixed` — фиксированное положение

Положение блока рассчитывается в соответствии с моделью абсолютного позиционирования, а затем он фиксируется относительно области просмотра или страницы. Два объявления могут быть отделены друг от друга с помощью правила `@media`, как это показано в примере:

```
@media screen { H1#first { position: fixed; } }
@media print { H1#first { position: static; } }
```



Управляя позиционированием, можно различным образом размещать блоки информации на странице, вплоть до создания эффектов наложения, перетекания, градиента и т.п.

ЛИСТ
изменений рабочей учебной программы по дисциплине
ОП.12 «Web-программирование»

Дополнения и изменения, вносимые в рабочую программу дисциплины на 2016-2017 уч. г.

Основания внесения дополнений и изменений	Раздел РПД, в который вносятся изменения*	Содержание вносимых дополнений, изменений*
Предложение работодателя	нет	нет
Предложение составителя программы	нет	нет
Приобретение, издание литературы, обновление перечня и содержания ЭБС, баз данных	Разделы 2.4.6 и 5 Перечень основной и дополнительной учебной литературы	Обновлен список рекомендуемой литературы

Составитель: канд. физ.-мат. наук, доцент



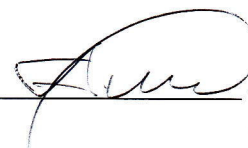
А. Н. Чернышев,

И.О. Трепалин

Утвержден на заседании предметно-цикловой комиссии *физико-математических дисциплин и специальных дисциплин специальности Компьютерные сети*,

протокол №1 от 29 августа 2016 г.

Председатель предметной (цикловой) комиссии
физико-математических дисциплин и специальных
дисциплин специальности Компьютерные сети



А. Б. Шишкин
29 августа 2016 г.

Начальник УМО филиала



А. В. Баранов
29 августа 2016 г.

Заведующая библиотекой филиала



М. В. Фуфалько
29 августа 2016 г.

Рецензия
на рабочую программу учебной дисциплины
ОП.12 «Web-программирование»
для специальности 09.02.02 Компьютерные сети

Рабочая программа учебной дисциплины ОП.12 «Web-программирование» разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования 09.02.02 Компьютерные сети, утвержденного приказом Минобрнауки РФ от 28.07.2014 №803 (зарегистрирован в Минюсте России 20.08.2014 № 33713). Рабочая программа является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.02 Компьютерные сети с квалификацией «Техник по компьютерным сетям». Учебная дисциплина ОП.12 изучается в составе профессионального учебного цикла ОП «Общепрофессиональные дисциплины» учебного плана ОПОП СПО. Обучение проводится на базе основного общего образования и нацелено на получение среднего общего образования. Рабочая программа составлена для очной формы обучения.

Рабочая программа учебной дисциплины составлена логично, структура дисциплины соответствует принципу единства теоретического и практического обучения, разделы выделены дидактически целесообразно. Последовательность тем, предлагаемых к изучению, направлена на качественное усвоение учебного материала. Виды самостоятельных работ позволяют обобщить и углубить изучаемый материал и направлены на закрепление умения поиска, накопления и обработки информации. Система знаний и умений, заложенная в содержании, обеспечивает освоение общих и профессиональных компетенций.

Паспорт программы обоснованно и полно отражает содержание дисциплины, ее роль и место в подготовке специалиста среднего звена, раскрывает цели и задачи учебной дисциплины. Определены требования к умениям и знаниям студентов. Программа рассчитана на изучение в 7 семестре в объеме 75 часов. Тематический план и содержание учебной дисциплины раскрывает последовательность прохождения тем, соответствует тематическому плану и распределению часов. В программе определены форма проведения, цели, задачи учебной дисциплины, представлены обязательные формы отчетности. В программе реализованы дидактические принципы обучения: целостность, структурность; отражена взаимосвязь между отдельными элементами структуры.

Содержание рабочей программы учебной дисциплины полностью соответствует требованиям Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.02 Компьютерные сети. Анализ раздела «Условия реализации программы учебной дисциплины», позволяет сделать вывод, что образовательное учреждение располагает материально-технической базой, отвечающей современным требованиям подготовки специалистов, обеспечивает проведение всех видов лабораторных и практических занятий, междисциплинарной подготовки, учебной практики, предусмотренных программой. Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники, изданные в последнее время (не позднее 5 лет). Перечисленные Интернет-ресурсы актуальны и достоверны.

Разработанная программа учебной дисциплины ОП.12 «Web-программирование» может быть рекомендована для использования в учебном процессе при подготовке по специальности 09.02.02 Компьютерные сети.

Начальник отдела информационных технологий ОАО «Сад-Гигант»



П.А. Дудник

Рецензия
на рабочую программу учебной дисциплины
ОП.12 «Web-программирование»
для специальности 09.02.02 Компьютерные сети

Рабочая программа учебной дисциплины ОП.12 «Web-программирование» разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования 09.02.02 Компьютерные сети, утвержденного приказом Минобрнауки РФ от 28.07.2014 №803 (зарегистрирован в Минюсте России 20.08.2014 № 33713).

В результате освоения дисциплины обучающийся должен уметь: использовать графические редакторы для обработки изображений, размещаемых на web-сайте; использовать язык гипертекстовой разметки HTML для создания web-страниц; создавать динамические web-страницы с использованием JavaScript; использовать PHP для создания web-страниц; осуществлять доступ к базам данных при проектировании web-сайта; настраивать конфигурацию web-сервера.

Перечисленные умения предполагают, что студент должен освоить следующие знания: методы проектирования web-сайта как статичной информационной системы; методы проектирования web-сайта как динамичной информационной системы; теорию использования графики на web-страницах; методы обработки и редактирования цифровых изображений; программные средства стороны клиента, используемые для создания web-страниц; программные средства стороны сервера, используемые для создания web-страниц; программные средства для создания баз данных; программные средства создания web-сервера; основные принципы конфигурации реального web-сервера; программные средства, используемые для размещения и сопровождения web-страниц.

В результате изучения дисциплины обучающийся должен приобрести практический опыт: проектирования web-сайта; оптимизации изображений для размещения на web-сайте; создания web-сайта средствами программирования на стороне клиента; проектирования web-сайта на стороне сервера; создания баз данных на стороне сервера; размещения web-сайта на сервере.

Рабочая программа рассчитана на изучение в 7 семестре в объеме 75 часов (50 часов – обязательная учебная нагрузка, 25 часов – самостоятельная работа, дифференцированный зачет). Освоение учебной дисциплины включает изучение следующих разделов:

- Раздел 1. HTML и CSS. Основы создания сайта.
- Раздел 2. Клиентское web-программирование. JavaScript.
- Раздел 3. Серверное web-программирование. PHP и MySQL.

Реализация программы дисциплины требует наличия лаборатории программного обеспечения компьютерных сетей, программирования и баз данных. Дисциплина нацелена на формирование профессиональных компетенций (ПК 2.2, 2.3, 3.1). Обучение студентов осуществляется по традиционной технологии с включением инновационных элементов: аудиовизуальные технологии, технология проблемного обучения, технология работы в малых группах.

Содержание рабочей программы учебной дисциплины полностью соответствует требованиям Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.02 Компьютерные сети. Изучение данной дисциплины способствует эффективной и качественной подготовке молодых специалистов в области Web-программирования.

Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники, изданные в последнее время (не позднее 5 лет). Перечисленные Интернет-ресурсы актуальны и достоверны.

Разработанная программа учебной дисциплины ОП.12 «Web-программирование» может быть рекомендована для использования в учебном процессе при подготовке по специальности 09.02.02 Компьютерные сети.

Рецензент, инженер-программист 1 категории,
отдел УСУТП управление АСУТП, КИПиА,
МОП Краснодарского РПУ филиала
«Макрорегион ЮГ» ООО ИК «Сибинтек»

 **ООО ИК
«СИБИНТЕК»
КРПУ АСУТП**

М.В. Литус