



1920

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
филиал федерального государственного бюджетного образовательного учреждения
высшего образования Кубанский государственный университет»
в г.Геленджике



директор по работе с филиалами

А.А. Евдокимов

2020 г.

Рабочая программа дисциплины
ОП.12 ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ. WEB
ПРОГРАММИРОВАНИЕ
специальность 09.02.03 Программирование в компьютерных системах

2020

Рабочая программа учебной дисциплины ОП.12 Интернет программирование. Web программирование разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее СПО) 09.02.03 Программирование в компьютерных системах, утвержденного приказом Минобрнауки РФ от 28.07.2014 №804 (зарегистрирован в Минюсте России 21.08.2014 № 33733)

Дисциплина	ОП.12 ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ. WEB ПРОГРАММИРОВАНИЕ	
Форма обучения	очная	
Учебный год	2020-2021	
4 курс		7 семестр
лекции		20 час.
практические занятия		40 час.
самостоятельные занятия		30 час.
форма итогового контроля		диф.зачет

Составитель: преподаватель



Т.П. Кривошеенко

профессиональных дисциплин специальности Программирование в компьютерных системах

Протокол № 10 от «27» мая 2020 г.

Председатель предметной (цикловой) комиссии

профессиональных дисциплин специальности

09.02.03 Программирование в компьютерных системах



Л.А. Благова

подпись

Директор ООО «Современные
Информационные технологии»

А.В.Сметанин

Программист ГБУЗ

"Геленджикский психоневрологический диспансер"
министерства здравоохранения Краснодарского края

Е.В.Мельников

ЛИСТ
согласования рабочей учебной программы по дисциплине
ОП.12 ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ. WEB
ПРОГРАММИРОВАНИЕ
Специальность среднего профессионального образования:
09.02.03 Программирование в компьютерных системах

СОГЛАСОВАНО:

Зам. директора по УР филиала



Т. А. Резуненко

«27» мая 2020 г.

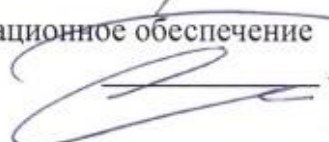
Заведующая сектором библиотеки



Л. Г. Соколова

«27» мая 2020 г.

Инженер-электроник (программно-информационное обеспечение
образовательной программы)



А. В. Сметанин

«27» мая 2020 г.

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.12	
Интернет программирование. Web программирование	3
1.1. Область применения программы	3
1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:	3
1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины:	4
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)	5
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	11
2.1. Объем учебной дисциплины и виды учебной работы	11
2.2. Структура дисциплины	11
2.3. Тематический план и содержание учебной дисциплины	11
2.4. Содержание разделов дисциплины	13
2.4.1. Занятия лекционного типа	13
2.4.2. Занятия семинарского типа	17
2.4.3. Практические занятия (Лабораторные занятия)	17
2.4.4. Содержание самостоятельной работы	17
Примерная тематика рефератов:	17
Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	18
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	23
4.2. Перечень необходимого программного обеспечения	23
5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	24
6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ	26

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ	30
7.1. Паспорт фонда оценочных средств	30
7.2. Критерии оценки знаний	30
7.3. Оценочные средств для проведения текущей аттестации.....	31
7.4. Оценочные средств для проведения промежуточной аттестации.....	33
7.4.1. Примерные вопросы для проведения промежуточной аттестации (экзамен, зачет, диф. зачет).....	34
7.4.2. Примерные экзаменационные задачи на экзамен/диф зачет.....	35
8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	36

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.12 Интернет программирование. Web программирование

1.1. Область применения программы

Рабочая программа учебной дисциплины ОП.12 Интернет программирование. Web программирование является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах.

1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:

Учебная дисциплина относится к общепрофессиональным дисциплинам профессионального цикла ОП.

Для освоения дисциплины студенты используют знания, умения и навыки, сформированные при изучении дисциплины ПД.02 «Информатика», а также ОП.02 «Архитектура компьютерных систем» (ПК3.1, ПК 3.2, ПК 3.4), ОП.04 «Интернет программирование. Web программирование» (ПК3.1, ПК 3.2, ПК 3.4)

№ п. п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	практический опыт (владеть)
1	ПК 3.1	Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения	Этапы решения задачи на компьютере	Работать в среде программирования	Владеть навыками анализа проектной и технической документации на уровне взаимодействия компонент программного обеспечения
2		Выполнять интеграцию модулей в программную систему	Принципы управления ресурсами в операционной системе, Особенности построения и функционирования операционных систем Unix и Windows	Управлять дисками и файловыми системами	Иметь практический опыт интеграции модулей в программную систему
3	ПК 3.4	Осуществлять разработку текстовых наборов и текстовых сценариев	Основные компоненты программного обеспечения компьютерных систем	Получать информацию о параметрах компьютерной системы	Иметь практический опыт получения информации о параметрах компьютерной системы

4	ПК 3.1	Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения	Методы разработки технических документов, методические и нормативные документы, техническую документацию, а также предложения и мероприятия по реализации разработанных проектов и программ	Разработка технических заданий: способность осуществлять разработку технических заданий проектов, технико-экономического обоснования	Проверять соответствие выполняемых работ требованиям проектной документации, анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения
	ПК 3.2	Выполнять интеграцию модулей в программную систему	Знать методы и технологии разработки программной системы, способы генерации исполняемого кода	Уметь выполнять интеграцию модулей в программную систему	Владеть методами и инструментами анализа и проектирования программного обеспечения, технологией интеграции проектных компонент и подсистем и поддерживающим инструментарием
	ПК 3.4	Осуществлять разработку тестовых наборов и тестовых сценариев	Знать методы и средства разработки тестовых сценариев и тестового кода	Уметь разрабатывать тестовые сценарии по спецификациям требований	Разрабатывать рекомендации по коррективам программных систем в процессе разработки, тестирования и внедрения программной системы

1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины:

В результате изучения обязательной части учебного цикла дисциплины обучающийся должен: **уметь:**

- использовать графические редакторы для обработки изображения, 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;

знать:

- методы проектирования web-сайта как статичной информационной системы;
- методы проектирования web-сайта как динамической 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.

Максимальная учебная нагрузка обучающегося 90 час., в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 60 часов;
- самостоятельная работа обучающегося 30 часов.

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

Обучающийся должен обладать общими компетенциями, включающими в себя способность:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности.

ОК 6. Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

ПК 3.2. Выполнять интеграцию модулей в программную систему.

ПК 3.3 Выполнять отладку программного продукта с использованием специальных программных средств.

ПК 3.4. Осуществлять разработку тестовых наборов и тестовых сценариев.

ПК 3.5 Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	практический опыт (владеть)
1.	ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	свободное владение профессиональным и знаниями в области web-программирования, использование современных компьютерных программных продуктов для решения задач профессиональной деятельности и за ее пределами	умения использовать знания в области интернет программирования для развития профессиональных навыков, способность использовать Интернет программирование. WEB программирование и современные инструментальные программные средства при решении социальных и профессиональных задач	Практическое использование профессиональных знаний: способность самостоятельно использовать Интернет программирование в предметной области и смежных отраслях, использование на практике интегрированных знаний в области web-программирования, умение выдвигать и применять идеи, вносить оригинальный вклад в будущую профессию
2.	ОК 2	Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество	знание основных тенденций развития web-программирования, способность использовать их базовые положения при решении социальных и профессиональных задач, развивать способность к приращению	умение использовать Интернет программирование для решения различных социальных, производственных, управленческих и других профессиональных задач,	способностью критически переосмысливать накопленный опыт, вносить изменения в рабочие процессы с учетом инноваций, совершенствовать навыки самостоятельной разработки методов и средств автоматизации информационных

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	практический опыт (владеть)
			знаний и внедрению передового российского и мирового опыта в своей профессиональной и междисциплинарной областях	критически оценить освоенные технологии, границы их применимости	процессов; делать свой вклад в оптимизацию рабочих процессов с учетом развития науки и технологий
3.	ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	Знать методы принятия решений в рамках компетентности специалиста, знать о кризис-менеджменте	Анализ информации: способность осуществлять сбор, обработку, анализ и систематизацию информации, обобщать и критически оценивать результаты	Выработка и принятие управленческих решений: способность разрабатывать варианты управленческих решений и обосновывать их выбор в рамках компетентности специалиста. Кризис-менеджмент: способностью управлять в кризисных ситуациях и применять технологии кризис-менеджмента.
4.	ОК 4	Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития	Знание программного обеспечения: способность выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения, понимание концепций и атрибутов качества программного обеспечения (надежности, безопасности, удобства использования), в том числе, роли людей, процессов, методов, инструментов и технологий обеспечения	Работать с нормативно-технической документацией: способностью осуществлять подбор, изучение, анализ и обобщение нормативных и методических материалов по профилю деятельности из электронных библиотек, реферативных журналов, сети Интернет, способность использовать нормативные правовые документы в своей профессиональн	способность осуществлять сбор, обработку, анализ и систематизацию необходимой информации, выбор подходящей технологии, инструментальных средств решения профессиональных задач, используя обзоры научной литературы и электронные информационно-образовательные ресурсы, информационно-коммуникационные технологии

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	практический опыт (владеть)
			качества	ой деятельности	
5.	ОК 5	Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности	владение теоретическими основами информационных технологий, готовность применять основные методы ИТ в своей профессиональной деятельности, знание методов самостоятельного поиска и использования различных источников информации по проблеме	осуществлять сбор, обработку, анализ и систематизацию научно-технической информации по профессии, выбирать методику и средства решения задач, используя научную литературу и электронные информационно-образовательные ресурсы, информационно-коммуникационные технологии	способность учитывать современные тенденции развития прикладной математики, информатики и вычислительной техники, компьютерных технологий в профессиональной деятельности, способность проводить научные, в том числе маркетинговые, исследования; готовность разрабатывать необходимое программное обеспечение для совершенствования профессиональной деятельности
6.	ОК 6	Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями	знание методов работы в коллективе и команде; понимание психологических особенностей функционирования личности, группы, общества, мирового сообщества	способность включаться в работу профессиональных групп; способность организовывать сотрудничество обучающихся, поддерживать активность и инициативность, самостоятельность и творческие способности, готовность к взаимодействию с учениками, родителями, коллегами, социальными партнерами	способность использовать современные информационно-коммуникативные технологии в работе с текстами, информацией, результатами исследований и для реализации профессиональной деятельности; способность делать вклад в личностный рост и повышение эффективности других участников профессиональной деятельности
7.	ОК 7	Брать на себя ответственность за работу членов	Владеть психологическими знаниями методов	Уметь развивать аналитическое	Развивать в себе аналитическое мышление,

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	практический опыт (владеть)
		команды (подчиненных), за результат выполнения заданий.	работы в коллективе и команде; понимать психологических особенностей функционирования личности, группы, общества, мирового сообщества	мышление, ответственность, коммуникабельность, креативность, инициативность, эмоциональную сдержанность, лидерские и организаторские качества	методичность, дисциплинированность, коммуникабельность, креативность, организованность, инициативность, стрессоустойчивость, толерантность, ответственность, требовательность, коммуникабельность, умение убеждать
8.	ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	Следить за отечественными и зарубежными разработками в области информационных и коммуникационных технологий	Осваивать новые методы и технологии в области информационных технологий	Развивать способность к освоению новых технологий; организовывать самообразование и повышение своей квалификации
9.	ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	Работа со знаниями: способность к приращению знаний и внедрению передового российского и мирового опыта в своей профессиональной и междисциплинарной областях	Уметь выполнять разнообразные профессиональные задачи с использованием на практике своих знаний и навыков в различных меняющихся и нестандартных условиях	способность и готовность к творческой адаптации к конкретным условиям выполняемых задач и их инновационным решениям, способность порождать новые идеи в условиях смены технологий
10.	ПК 3.1	Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения	методы проектирования web-сайта как статичной информационной системы и как динамической информационной системы	использовать графические редакторы для обработки изображений, размещаемых на web-сайте	проектирование web-сайта, оптимизации изображений для размещения на web-сайте
11.	ПК 3.2	Выполнять интеграцию модулей в программную систему	программные средства для создания баз данных и web-сервера	создавать динамические web-страницы с использованием JavaScript	создания web-сайта средствами программирования на стороне клиента

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	практический опыт (владеть)
12.	ПК 3.3	Выполнять отладку программного продукта с использованием специализированных программных средств	программные средства стороны клиента и стороны сервера, используемые для создания web-страниц, программные средства используемые для размещения и сопровождения web-страниц	использовать PHP для создания web-страниц	проектирования web-сайта на стороне сервера
13.	ПК 3.4	Осуществлять разработку тестовых наборов и тестовых сценариев	теорию использования графики на web-страницах, методы обработки и редактирования цифровых изображений	использовать язык гипертекстовой разметки HTML для создания web-страниц	создания баз данных на стороне сервера
14.	ПК 3.5	Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования	основные принципы конфигурации реального web-сервера	осуществлять доступ к базам данных при проектировании web-сайта, настраивать конфигурацию web-сервера	размещения web-сайта на сервере

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	90
Обязательная аудиторная учебная нагрузка (всего)	60
в том числе:	
занятия лекционного типа	20
практические занятия	40
Самостоятельная работа обучающегося (всего)	30
в том числе:	
самостоятельная внеаудиторная работа в виде домашних практических заданий, индивидуальных заданий, самостоятельного подбора и изучения дополнительного теоретического материала	30
Промежуточная аттестация в форме зачета/экзамена/дифзачета	Диф.зачет

2.2. Структура дисциплины

Наименование разделов и тем	Количество аудиторных часов			Самостоятельная работа обучающегося (час)
	Всего	Теоретическое обучение	Практические и лабораторные занятия	
Раздел 1. HTML и CSS. Основы создания сайта.		4	4	
Раздел 2. Клиентское web-программирование. JavaScript		6	10	
Раздел 3. Серверное web-программирование. PHP и MySQL		10	26	
Всего по дисциплине		20	40	30

2.3. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, практические работы, самостоятельная работа обучающихся, курсовая работа (если предусмотрена)	Объем часов	Уровень освоения
1	2	3	4
Раздел 1. HTML и CSS. Основы создания сайта.	Содержание учебного материала		
	Лекции		
	Лекция 1. Основы HTML. Каскадные таблицы (CSS).	2	1,2
	Лекция 2. Использование таблиц для верстки. Использование слоев в HTML. Верстка при помощи слоев.	2	1,2
	Практические (лабораторные) занятия	2	
	Практическое занятие 1. Основы HTML. Каскадные таблицы (CSS).		
	Практическое занятие 2. Использование таблиц для верстки. Верстка при помощи слоев	2	
Самостоятельная работа обучающихся	10		
Раздел 2.	Содержание учебного материала		
	Лекции		

Клиентское web-программирование. JavaScript	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		
	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»	6	
	Самостоятельная работа обучающихся	10	
Раздел 3. Серверное web-программирование. PHP и MySQL	Содержание учебного материала		
	Лекции		
	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 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		
	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 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	
	Практическое занятие 14. Использование Cookie. Почтовые рассылки в PHP.	4	
	Практическое занятие 15. Работа над проектом «Расширение функциональности сайта средствами PHP и MySQL»	6	
	Самостоятельная работа обучающихся	10	
	Всего:	90	

Для характеристики уровня освоения учебного материала используются следующие обозначения: 1. – ознакомительный (узнавание ранее изученных объектов, свойств); 2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством) 3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

2.4. Содержание разделов дисциплины

2.4.1. Занятия лекционного типа

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
<i>7 семестр</i>			
Раздел 1. HTML и CSS. Основы создания сайта.			
1	Основы HTML.	<ul style="list-style-type: none">• Структура HTML страницы: заголовков и тело документа.• Использование в HTML комментариев и спецсимволов.• Теги и оформление текста.• Работа со стилем текста, его разбиение на блоки.• Гиперссылки, их объявление на странице.• Иллюстрации на веб-странице и их правильное использование.• Обмен с сервером. Формы и элементы управления.	У
2	Каскадные таблицы (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.	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 вёрстки.• Заполнение разметки частями нарезанного макета.• Методы размещения элементов при помощи слоев.	У
Раздел 2. Клиентское web-программирование. JavaScript			
5	Введение в JavaScript. Способы внедрения JS, стандартные	<ul style="list-style-type: none">• История JavaScript.• Понятие JavaScript.• Область применения JavaScript.• Литералы, переменные, выражения.	У, Р

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
	функции.	<ul style="list-style-type: none"> • JavaScript-код на странице. • Подключение внешнего JavaScript файла. • Знакомство с alert() и console(). • Самые распространенные ошибки JavaScript. 	
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. 	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. 	У
8	Работа с формами, валидация.	<ul style="list-style-type: none"> • Получение параметра ввода пользователя. • Изменения параметра ввода, преобразование. • Валидация. 	У
9	Классы и прототипы.	<ul style="list-style-type: none"> • Конструкторы. • Прототип и наследование свойств. • Создание класса, Обращение к методам класса, удаление класса. 	У
10	Cookie.	<ul style="list-style-type: none"> • История Cookie. • Безопасность и ограничения. • Область применения Cookie. • Чтение Cookie. Удаление Cookie. 	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 связи 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. 	У

№ раздела	Наименование раздела	Содержание раздела	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. 	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 при работе с массивом. • Функции для работы с массивами • Многомерные массивы: отличие от одномерных и работа с ними в PHP. • Предопределенные массивы. 	У
14	Запросы HTTP, параметры URL и формы HTML.	<ul style="list-style-type: none"> • Принципы работы протокола HTTP. • Запросы GET и POST. • Метод GET. Передача пользовательских параметров через URL. Параметры запроса URL. • Обработка параметров, полученных методом GET и PHP. • Отправка параметров методом POST. Обработка отправки запросов через HTML форму. • Преимущества и недостатки методов GET и POST. 	У
15	Стандартные средства PHP.	<ul style="list-style-type: none"> • Документация по PHP и сети Интернет. • 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. 	У

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
16	Использование файлов PHP.	<ul style="list-style-type: none"> • Принципы работы с файлами в PHP. • Режим 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. • Работа с каталогами, расположенными на сервере. • Получение списка файлов и директорий. 	У
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. 	У
18	Взаимодействие PHP и MySQL.	<ul style="list-style-type: none"> • Расширения языка PHP и MySQL. • Соединение с сервером баз данных. • Запросы для получения информации. • Обработка результата запроса. • Обеспечение безопасности данных. Борьба с SQL-инъекциями. 	У
19	Авторизация пользователей. Сессии.	<ul style="list-style-type: none"> • Использование сессий. • Создание сессий. • Хранение данных о пользователе. • Хранение паролей в БД. Хеширование и алгоритм md5. • Форма регистрации пользователя. Сохранение профиля. • Авторизация пользователя. • Чтение данных сессии. • Вывод пользователя с сайта. Закрытие сессии. 	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. • Работа с Cookie :создание, чтение, удаление. • Как работает электронная почта. • Расширения языка PHP для работы с электронной почтой. • Создание электронного письма средствами PHP. 	У
Примечание: Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа			

2.4.2. Занятия семинарского типа

– не предусмотрены

2.4.3. Практические занятия

№	Наименование раздела	Наименование практических (лабораторных) работ	Форма текущего контроля
<i>1 семестр</i>			
1	2	3	4
1.	Раздел 1. HTML и CSS. Основы создания сайта.	Практическое занятие 1. Основы HTML. Каскадные таблицы (CSS).	ПР
		Практическое занятие 2. Использование таблиц для верстки. Верстка при помощи слоев	
2.	Раздел 2. Клиентское web-программирование. JavaScript	Практическое занятие 3. Способы внедрения JS, стандартные функции. Работа с переменными и операторами сравнения.	ПР
		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»	
3.	Раздел 3. Серверное web-программирование. PHP и MySQL	Практическое занятие 6. Основы PHP. Условные операторы. Понятие функции.	ПР
		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 PHP.	
		Практическое занятие 11. СУБД MySQL. Разработка базы 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»		

Примечание: ПР- практическая работа, ЛР- лабораторная работа; Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа

2.4.4. 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. Общие понятия языка разметки гипертекста- HTML.
4. Важность безопасности в веб-приложениях.

5. Особенности языка PHP.
6. Связь PHP и HTML.
7. Основные правила веб-дизайна.
8. Аутентификация пользователей средствами PHP.
9. Работа с датой и временем в PHP.
10. Создание HTML-страницы средствами PHP.
11. Обзор технологий, используемых для обеспечения безопасности.

Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа обучающихся является важнейшей формой учебно-познавательного процесса.

Основная цель самостоятельной работы обучающегося при изучении дисциплины – закрепить теоретические знания, полученные в ход лекционных занятий, а также сформировать практические навыки подготовки в области информационных технологий.

Самостоятельная работа обучающегося в процессе освоения дисциплины включает:

- изучение основной и дополнительной литературы по курсу;
- самостоятельное изучение некоторых вопросов (конспектирование);
- работу с электронными учебными ресурсами;
- изучение материалов периодической печати, интернет ресурсов;
- подготовку к тестированию;
- подготовку к практическим (лабораторным) занятиям,
- самостоятельное выполнение домашних заданий,
- подготовку реферата (доклада, эссе) по одной из проблем курса.

На самостоятельную работу обучающихся отводится 30 часов учебного времени.

№	Наименование раздела, темы, вида СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Раздел 1. HTML и CSS. Основы создания сайта.	1. Гвоздева, В.А. Основы построения автоматизированных информационных систем: учебник для СПО/В.А. Гвоздева, И.Ю. Лаврентьева. -М.:Форум:Инфра-М,2013.-317с. 2. Фуфаев, Э.В. Разработка и эксплуатация удаленных баз данных: учебник для СПО/Э.В. Фуфаев, Д.Е. Фуфаев.-М.:Академия,2014.-250 с. 3. Фуфаев, Э.В. Разработка и эксплуатация удаленных баз данных : учебник для СПО/Э.В. Фуфаев, Д.Е. Фуфаев.-М.:Академия,2012.-250 с.
2.	Раздел 2. Клиентское web-программирование. JavaScript	1. Гасанов, Э.В. Практикум по созданию интернет-проектов. Основы языка программирования PHP. Ч.1 [Электронный ресурс]: учебное пособие / Э.В. Гасанов и др.- М.: Книгодел, 2013.- 160 с.- URL: http://biblioclub.ru/index.php?page=book&id=230535&sr=1

		<p>2. Громов, Ю.Ю. Основы Web-инжиниринга [Электронный ресурс]: учебное пособие / Ю.Ю. Громов и др.- Тамбов: ТГТУ, 2012.- 240 с. - URL: http://biblioclub.ru/index.php?page=book&id=277648&sr=1</p> <p>3. Гениатулина, Е.В. CMS- системы управления контентом [Электронный ресурс]: учебное пособие /Е.В. Гениатулина.- Новосибирск: НГТУ, 2015.- 63 с.- URL: http://biblioclub.ru/index.php?page=book&id=438332&sr=1</p>
3	Раздел 3. Серверное web-программирование. PHP и MySQL	<p>1. Гвоздева, В.А. Основы построения автоматизированных информационных систем: учебник для СПО/В.А. Гвоздева, И.Ю. Лаврентьева. -М.:Форум:Инфра-М,2013.-317с.</p> <p>2. Фуфаев, Э.В. Разработка и эксплуатация удаленных баз данных: учебник для СПО/Э.В. Фуфаев, Д.Е. Фуфаев.-М.:Академия,2014.-250 с.</p> <p>3. Фуфаев, Э.В. Разработка и эксплуатация удаленных баз данных : учебник для СПО/Э.В. Фуфаев, Д.Е. Фуфаев.-М.:Академия,2012.-250 с.</p>

Кроме перечисленных источников по темам самостоятельной работы, студент может воспользоваться Электронно-библиотечными системами (ЭБС), профессиональными базами данных, электронными базами периодических изданий, другими информационными ресурсами, указанными в разделе 5.4 «Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины», включающий ресурсы, доступ к которым обеспечен по договорам с правообладателями, и образовательные, научные, справочные ресурсы открытого доступа, имеющие статус официальных (федеральные, отраслевые, учреждений, организаций и т.п.), а также поисковыми системами сети Интернет для поиска и работы с необходимой информацией.

Для освоения данной дисциплины и выполнения предусмотренных учебной программой курса заданий по самостоятельной работе обучающийся может использовать учебно-методическое обеспечение:

-методические рекомендации преподавателя к практическим (лабораторным) занятиям;

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. Обучающийся должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Обучающиеся 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.

3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для реализации компетентностного подхода предусматривается использование в учебном процессе 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. Обязательны компьютерные лабораторные практикумы по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное тестирование, тематические презентации, интерактивные технологии.

3.1. Образовательные технологии при проведении лекций

№	Тема	Виды применяемых образовательных технологий	Кол-во час
1	2	3	4
Раздел 1. HTML и CSS. Основы создания сайта.			
1	Основы HTML. Каскадные таблицы (CSS).	Аудиовизуальная технология Технология проблемного обучения	2*
2	Использование таблиц для верстки. Использование слоев в HTML. Верстка при помощи слоев.	Аудиовизуальная технология Технология проблемного обучения	2*
Раздел 2. Клиентское web-программирование. JavaScript			
3	Введение в JavaScript. Способы внедрения JS, стандартные функции. Типы переменных. Работа с переменными и операторами сравнения.	Аудиовизуальная технология Технология проблемного обучения	2*
4	Управление web-объектами. Знакомство с событиями. Работа с формами, валидация.	Аудиовизуальная технология Технология проблемного обучения	2*
5	Классы и прототипы. Cookies.	Аудиовизуальная технология Технология проблемного обучения	2*
Раздел 3. Серверное web-программирование. PHP и MySQL			
6	Основы PHP. Условные операторы. Понятие функции.	Аудиовизуальная технология Технология проблемного обучения	2*
7	Циклы и массивы. Запросы HTTP, параметры URL и формы HTML.	Аудиовизуальная технология Технология проблемного обучения	2*
8	Стандартные средства PHP. Использование файлов PHP.	Аудиовизуальная технология Технология проблемного обучения	2*
9	СУБД MySQL. Разработка базы данных web-приложения. Взаимодействие PHP и MySQL.	Аудиовизуальная технология Технология проблемного обучения	2*

10	Авторизация пользователей. Сессии. Использование Cookie. Почтовые рассылки в PHP.	Аудиовизуальная технология Технология проблемного обучения	2*
Итого по курсу			20
в том числе интерактивное обучение*			20*

3.2. Образовательные технологии при проведении практических занятий (лабораторных работ)

№	Тема занятия	Виды применяемых образовательных технологий	Кол. час
1	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1 раздела 1 «. Основы HTML. Каскадные таблицы (CSS).»	Выполнение практической работы	2
2	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2 раздела 1 «Использование таблиц для верстки. Верстка при помощи слоев»	Выполнение практической работы	2
3	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3 раздела 2 «Способы внедрения JS, стандартные функции. Работа с переменными и операторами сравнения.»	Выполнение практической работы	2*
4	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4 раздела 2 «Управление web-объектами. Знакомство с событиями. Работа с формами, валидация.»	Выполнение практической работы	2*
5	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5 раздела 2 «Работа над проектом «Разработка сайта средствами HTML и JS»»	Выполнение практической работы	6*
6	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6 раздела 3 «Основы PHP. Условные операторы. Понятие функции»	Выполнение практической работы	2*
7	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7 раздела 3 «Циклы и массивы.»	Выполнение практической работы	2*
8	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8 раздела 3 «Запросы HTTP, параметры URL и формы HTML.»	Выполнение практической работы	2*
9	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 9 раздела 3 «Стандартные средства PHP.»	Выполнение практической работы	2*
10	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №10 раздела 3 «Использование файлов PHP.»	Выполнение практической работы	2*
11	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №11 раздела 3 «СУБД MySQL. Разработка базы данных web-приложения»	Выполнение практической работы	2*
12	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №12 раздела 3 «Взаимодействие PHP и MySQL.»	Выполнение практической работы	2*
13	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №13 раздела 3 «Авторизация пользователей. Сессии»	Выполнение практической работы	2*
14	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №14 раздела 3 «Использование Cookie. Почтовые рассылки в	Выполнение практической работы	4*

	PHP»		
15	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №15 по раздела 3 «Расширение функциональности сайта средствами PHP и MySQL»	Выполнение практической работы	6*
		Итого по курсу	40
		в том числе интерактивное обучение*	40*

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

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

Реализация учебной дисциплины осуществляется в специально оборудованной лаборатории управления проектной деятельности

Оборудование Лаборатории управления проектной деятельностью:

- мультимедийный проектор, экран;
- персональные компьютеры;
- локальная компьютерная сеть с выходом в Интернет;
- учебная мебель;
- доска аудиторная.

Наглядные пособия:

Комплект плакатов

Электронные ресурсы

4.2. Перечень необходимого программного обеспечения

1. Microsoft Office Professional
2. PascalABC - визуальная среда программирования;
3. PascalABC.NET - визуальная среда программирования ;
4. WEB-среда разработки PascalABC.NET.
5. Среда разработки ПО PortableDelphi;
6. Разработчик инсталляторов InnoSetup;
7. 7-zip архиватор;
8. Adobe Acrobat Reader просмотрщик файлов ;
9. Adobe Flash Player –графический редактор;
10. Apache OpenOffice – офисный пакет;
11. FreeCommander - проводник;
12. Google Chrome - браузер;
13. LibreOffice – офисный пакет
14. Mozilla Firefox – браузер
15. nanoCAD версия 5.1 локальная
16. ЭкзамL – Система компьютерного тестирования;
17. Программный комплекс "Универсальный тест 4.0.0.1";
18. Lazarus – визуальная среда программирования;

5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ. WEB ПРОГРАММИРОВАНИЕ

5.1. Основная литература

1. Малашкевич, В.Б. Интернет-программирование : лабораторный практикум / В.Б. Малашкевич ; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с. : ил. - Библиогр.: с. 82. - ISBN 978-5-8158-1854-5 ; [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=476400
2. Черпаков, И. В. Основы программирования : учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2018. — 219 с. .- URL: <https://biblionline.ru/viewer/F79BE55A-C6F1-439D-9ED5-0D78A50B403F/osnovy-programmirovaniya#page/1>

5.2. Дополнительная литература

1. Немцова, Т.И. Компьютерная графика и web- дизайн: практикум: учебное пособие для студентов учреждений среднего профессионального образования/ Т.И. Немцова, Ю.В. Назарова; под ред. Л.Г. Гагариной.- Москва: ФОРУМ: ИНФРА-М, 2015.-287с.
2. Немцова, Т.И. Компьютерная графика и web- дизайн: практикум: учебное пособие для студентов учреждений среднего профессионального образования/ Т.И. Немцова, Ю.В. Назарова; под ред. Л.Г. Гагариной.- Москва: ФОРУМ: ИНФРА-М, 2014.- 399с.
3. Евсеев, Д.А. Web-дизайн в примерах и задачах : учебное пособие для вузов/Д.А. Евсеев, В.В. Трофимов; под ред. В.В. Трофимова.-М.:КноРус,2015.-263с.
4. Евсеев, Д.А. Web-дизайн в примерах и задачах : учебное пособие для вузов/Д.А. Евсеев, В.В. Трофимов; под ред. В.В. Трофимова.-М.:КноРус,2014.-263с.
5. Кузнецов, М.В. Самоучитель PHP 5/6 / М.В. Кузнецов, И.В. Симдянов.- Санкт-Петербург: БХВ-Петербург, 2014.-651с.
6. Дронов, В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов [Текст] / В. Дронов. - Санкт-Петербург : БХВ-Петербург, 2014. - 414 с.

5.3 Периодические издания

1. Среднее и профессиональное образование
2. Компьютер Пресс
3. Открытые системы.- URL: <http://biblioclub.ru/index.php?page=journal&jid=436083>
4. Информатика в школе .- URL: <http://dlib.eastview.com/browse/publication/18988>
5. Программные продукты и системы.- URL: <http://dlib.eastview.com/browse/publication/64086>
6. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
7. Системный администратор.- URL: <http://dlib.eastview.com/browse/publication/66751>
8. Computerword Россия.- URL: <http://dlib.eastview.com/browse/publication/64081>
9. Мир ПК.- URL: <http://dlib.eastview.com/browse/publication/64067>
10. Информационно-управляющие системы.- URL: <http://dlib.eastview.com/browse/publication/71235>
11. Журнал сетевых решений LAN.- URL: <http://dlib.eastview.com/browse/publication/64078>
12. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
13. Windows IT Pro/ Re.- URL: <http://biblioclub.ru/index.php?page=journal&jid=138741>

14. Прикладная информатика.- URL : http://elibrary.ru/title_about.asp?id=25599

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

1. ЭБС «Университетская библиотека ONLINE». – URL: www.biblioclub.ru
2. ЭБС издательства «Лань». – URL: <https://e.lanbook.com>
3. ЭБС «Юрайт». – URL: <http://www.biblio-online.ru/>
4. ЭБС «BOOK.ru» <https://www.book.ru>
5. ЭБС «ZNANIUM.COM» www.znanium.com
6. Электронный каталог Научной библиотеки КубГУ. –
[URL:http://212.192.134.46/MegaPro/Catalog/Home/Index](http://212.192.134.46/MegaPro/Catalog/Home/Index)
7. Электронная библиотека «Издательского дома «Гребенников» -
[URL:www.grebennikon.ru](http://www.grebennikon.ru)
8. Научная электронная библиотека (НЭБ) «eLibrary.ru». - [URL:http://www.elibrary.ru](http://www.elibrary.ru)
9. Базы данных компании «Ист Вью». - [URL:http://dlib.eastview.com](http://dlib.eastview.com)
10. Лекториум ТВ». - URL: <http://www.lektorium.tv/>
11. Национальная электронная библиотека «НЭБ». - [URL:http://нэб.рф/](http://нэб.рф/)
12. КиберЛенинка: научная электронная библиотека. – URL: <http://cyberleninka.ru/>
13. Единое окно доступа к образовательным ресурсам : федеральная ИС свободного доступа. – URL: <http://window.edu.ru>.
14. Справочно-правовая система «Консультант Плюс» - URL <http://www.consultant.ru>

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ «ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ. WEB ПРОГРАММИРОВАНИЕ»

Учащиеся 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. Это обеспечит более полную подготовку как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа учащихся является важнейшей формой учебно-познавательного процесса. Цель заданий для самостоятельной работы – закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

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. Учащийся должен изучить список нормативно-правовых актов и экономической литературы, рекомендуемый по учебной дисциплине; уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания учащийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

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

С точки зрения используемых методов лекции подразделяются следующим образом: информационно-объяснительная лекция, повествовательная, лекция-беседа, проблемная лекция и т. д.

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 «Интернет программирование. WEB программирование» проводятся в основном по схеме:

-устный опрос по теории в начале занятия (обсуждение теоретических проблемных вопросов по теме);

-работа в группах по разрешению различных ситуаций по теме занятия;

-решение практических задач индивидуально;

-подведение итогов занятия (или рефлексия);

-индивидуальные задания для подготовки к следующим 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 в г. Геленджике;

– 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.

Прочитав 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. 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 записка, 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 определённый 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 обращаться 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.

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.

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ

7.1. Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Раздел 1. HTML и CSS. Основы создания сайта.	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 3.1, ПК 3.3	Практическая работа
2.	Раздел 2. Клиентское web-программирование. JavaScript	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 3.4, ПК 3.5	Практическая работа
3.	Раздел 3. Серверное web-программирование. PHP и MySQL	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 3.2	Практическая работа

7.2. Критерии оценки знаний

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий, лабораторных работ, тестирования, а также выполнения обучающимися индивидуальных самостоятельных заданий.

Реферат. Продукт самостоятельной работы обучающегося, 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 привлеченных источников, глубине анализа проблемы, качестве обоснования авторской позиции, глубине раскрытия темы.

Тест. Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося. Тест оценивается по количеству правильных ответов (не менее 50%).

Критерии оценки знаний обучающихся в целом по дисциплине:

«отлично» - выставляется обучающемуся, 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 обучающемуся, 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 обучающемуся, 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 обучающемуся, 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 проводится в форме:

- фронтальный опрос
- индивидуальный устный опрос
- письменный контроль
- тестирование по теоретическому материалу
- практическая (лабораторная) работа
- защита реферата, эссе
- защита выполненного задания,

Форма аттестации	Знания	Умения	Практический опыт (владение)	Личные качества обучающегося	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературными источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Рефераты	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературными источниками	Оценка способности к самостоятельной работе и анализу литературных источников	Темы рефератов прилагаются
Практические (лабораторные) работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении практических задач.	Оценка навыков работы с вычислительной техникой, прикладными программными средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются

Тестирование	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков логического анализа и синтеза при сопоставлении конкретных понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются
--------------	---	--	---	--	---------------------

Примерные тестовые задания:

Не предусмотрены

Примерные вопросы для устного опроса (контрольных работ):

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.

Примерные вопросы для контроля самостоятельной работы:

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.

7.4. Оценочные средств для проведения промежуточной аттестации

Промежуточная аттестация

Форма аттестации	Знания	Умения	Практический опыт (владеть)	Личные качества обучающегося	Примеры оценочных средств
Экзамен/дифзачет	Контроль знания базовых положений в области информатики	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	Оценка навыков логического	Оценка способности	Задачи прилагаются

		задачи в области профессиональной деятельности	мышления при решении задач в области профессиональной деятельности	грамотно и четко излагать ход решения задач в области профессиональной деятельности и аргументировать результаты	
--	--	--	--	--	--

7.4.1. Примерные вопросы для проведения промежуточной аттестации (экзамен, зачет, диф. зачет)

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.

7.4.2. Примерные экзаменационные задачи на экзамен/диф зачет

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.

Другие оценочные средства по дисциплине не предусмотрены.

8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Приложение 1. Краткий конспект лекционных занятий

HTML-документ: первые шаги

HTML-документ – это файл с расширением *.html или *.htm, созданный с применением языка разметки гипертекста. Для форматирования документа (установки его параметров) служат теги – команды языка HTML. Общий вид записи тега выглядит так: `<Тег>. . . текст</Тег>`, где `<Тег>` является «открывающим» тегом (обозначает начало действия определенной команды в документе), а `</Тег>` – «закрывающий» тег (определяет место окончания действия). Пара тегов – закрывающий и открывающий – образуют элемент под названием «контейнер».

Главная информационная роль отводится **атрибутам** тегов. Атрибут является набором символов, значение которых соответствует значению определенного английского слова. У одного тега может быть несколько атрибутов.

Разберем конкретный пример. Тег `` применяется для форматирования текста. Однако он не способен указать подробные параметры форматирования (размер текста, шрифт и его начертание, цвет и другие). Зато это по силам атрибутам `face` (гарнитура шрифта), `color` (цвет), `dir` и `lang` (направление чтения текста и информация о языке), `size` (размер текста). HTML-код: ``, причем `font` – тег, `face` – атрибут, `Verdana, Arial . . .` – значение атрибута.

Теперь, обладая знаниями, достаточными на первое время, создадим наш первый html-документ. Для этого откройте любой текстовый или визуальный html-редактор (в последнем случае придется перейти в режим отображения кода). Начинающим веб-мастерам довольно часто советуют создавать первые веб-страницы с помощью обыкновенного текстового редактора (например, Блокнота в Windows). Трудно сказать, чем обусловлены подобные рекомендации, но для написания кода разумнее воспользоваться специализированными программными продуктами (EmEditor, Dreamweaver, HomeSite). Конечно, в начале будет трудно. Однако вскоре вы привыкните к выбранной программе, разберетесь в ее особенностях. А такие полезные функции как подсветка синтаксиса, сообщения об ошибках, встроенный браузер для наглядной демонстрации результатов, окупят все мучения.

Но мы немного отвлеклись. Итак, запустив html-редактор, введите следующий код:

```
<html>
<head>
<title>Моя страничка</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>
Мой первый документ
</body>
</html>
```

Рассмотрим форму записи кода. Любой документ начинается и заканчивается парой `<html>` и `</html>`. Хотя в данном случае теги набирались маленькими буквами, никакой ошибки не будет, если ввести теги заглавными буквами. Каждый тег располагается в отдельной строке. Это не правило, просто так комфортнее читать текст. Пара тегов `<head>` и `</head>` обозначает начало и конец заголовка документа. Метка `<title>` содержит служебную информацию о названии документа. Начало и конец тела документа - `<body>` и `</body>`.

Причем в теле документа находится его основное содержимое.

Подведем итоги:

Сегодня вы научились создавать простейшие html-документы и сделали первые шаги на пути к изучению профессиональной верстки веб-страниц.

Как не надо верстать сайт или 7 ошибок начинающего веб-мастера

Прежде чем приступить к продвижению сайта в сети Интернет, необходимо позаботиться об отсутствии в нем ошибок, связанных с неправильной версткой. Иначе любая, самая продуманная рекламная компания в поддержку веб-ресурса, окажется бесполезной тратой времени и денег.

Верстка сайта – процесс тяжелый и трудоемкий, особенно для начинающих веб-мастеров. Существует целый ряд наиболее часто встречающихся ошибок, устранением которых мы и займемся.

1. Некорректное использование JavaScript, апплетов, DHTML, сценариев CGI и SSI.

Если вы не слишком хорошо разбираетесь в технологиях **CGI** и **SSI**, поверхностно знаете языки **JavaScript** и **DHTML**, лучше на первых парах обойтись без них. Пусть ваш сайт не поражает использованием новейших технологий, зато у пользователей не возникнет проблем при просмотре страниц. А веб-браузеры не будут выдавать сообщения об ошибках. Следует также учитывать, что необоснованное использование апплетов приводит к увеличению размера страницы и времени ее загрузки. Поскольку в борьбе за постоянную аудиторию, скорость передачи данных играет решающую роль, старайтесь при проектировании страницы учитывать всевозможные способы доступа в Интернет (от стандартного Dial-Up до GPRS и широкополосного Интернета). Идеальным будет считаться вариант, если у пользователя Dial-Up доступа со скоростью, не превышающей 56 Кб/с страница вашего веб-ресурса загрузится в течение 30 секунд. Задача конечно не из легких, но кто сказал, что путь к успеху устлан розами?

Прежде чем приступить к продвижению сайта в сети Интернет, необходимо позаботиться об отсутствии в нем ошибок, связанных с неправильной версткой. Иначе любая, самая продуманная рекламная компания в поддержку веб-ресурса, окажется бесполезной тратой времени и денег.

Верстка сайта – процесс тяжелый и трудоемкий, особенно для начинающих веб-мастеров. Существует целый ряд наиболее часто встречающихся ошибок, устранением которых мы и займемся.

2. Графика и анимация на страницах веб-сайта. Чрезмерное применение.

Безусловно, красиво сделанная анимация и использование флеш-роликов оживляет сайт, придавая ему индивидуальный стиль. Но тут главное не перегнуть палку, в противном случае посетители, пришедшие за конкретной информацией, быстро устанут от беспорядочного мельтешения мультимедийных персонажей и уйдут к вашим конкурентам с более строгим дизайном и более скромными размерами загружаемых страниц.

Некоторые пользователи, в целях экономии трафика, просто-напросто отключают показ графических изображений. Чтобы было понятно, что находится на странице, прописывайте

комментарии (**alt**) к картинкам и баннерам. Тем более обязательны теги в том случае, если меню сайта целиком состоит из графических элементов.

3. Обилие рекламы.

В наше нелегкое время рыночной конкуренции сайты выживают за счет рекламы. Однако все же не стоит слишком явно демонстрировать свою заинтересованность в коммерческом успехе ресурса и размещать по два десятка баннеров на каждой странице. Это в прямом смысле отпугнет посетителей и понизит престиж сайта, особенно если вы рекламируете товары сомнительного качества, не вызывающие доверия (БАДЫ, сетевые пирамиды, способы разбогатеть ничего не делая и т.д.). Запомните, на одной странице допускается размещение не более двух баннеров больших размеров (один – вверху страницы, другой – внизу) и 4 баннера размера 88x31. Следите, чтобы они вписывались в общий дизайн и соответствовали тематике веб-ресурса.

4. Всплывающие окна.

Никогда не применяйте всплывающие окна без крайней необходимости. Опытные пользователи Интернета автоматически закрывают подобные окна, даже не просматривая их содержание. А большинство антивирусов блокируют всплывающие окна, подозревая их в наличие вредоносного кода.

Некоторые пользователи, в целях экономии трафика, просто-напросто отключают показ графических изображений. Чтобы было понятно, что находится на странице, прописывайте комментарии (**alt**) к картинкам и баннерам. Тем более обязательны теги в том случае, если меню сайта целиком состоит из графических элементов.

5. Дизайн сайта – «встречают по одежке, провожают по уму».

Применение несовместимых цветовых решений и прямо противоположных по стилю шрифтов, отсутствие каких-либо правил при оформлении текста, – и вот уже первое впечатление от вашего сайта, каким бы ни было его информационное наполнение, испорчено. Посетитель горестно вздыхает и уходит к конкурентам. Не слишком радостная картина, не правда ли?

Чтобы не оказаться в подобной ситуации, при работе над дизайном сайта, необходимо знать некоторые правила оформления:

- Цвета делятся на «теплые» и «холодные». К «теплым» относятся желтый, оранжевый, красный, малиновый и фиолетовый цвета. «Холодными» называют зеленый, голубой и синий цвета. Белый цвет считается нейтральным. Некоторые цвета в соседстве с другими кажутся нам или выступающими вперед, или отступающими назад. В основном выступающими воспринимаются теплые, светлые и насыщенные цвета, а отступающими – холодные, темные и ненасыщенные. Так что если необходимо что-то выделить, то можно поэкспериментировать с цветами.
- Если весь текст на странице, включая заголовки, выполнить одним шрифтом, то работать с таким текстом будет неудобно. Поэтому, чтобы документ легко читался, **заголовки обычно оформляются одним шрифтом, а основной текст – другим (меньшего размера)**. Полужирное начертание шрифта используется исключительно для заголовков. Курсивное начертание применяется и в заголовках и в основном

тексте (например, для выделения терминов).

Особое внимание уделите проверке орфографии и пунктуации. На сайте не должно быть грамматических ошибок, неправильно расставленных знаков препинания и опечаток.

7. Тестирование сайта на локальном сервере.

Это – не ошибка, более того, на заключительном этапе создания сайта тестирование на локальном сервере является обязательным. Это позволит увидеть, как будет выглядеть сайт в Интернете. Проведя тестирование, вы избежите многих ошибок, как крупных (см. пункт 1. «Некорректное использование JavaScript, апплетов, DHTML, сценариев CGI и SSI»), так и мелких ляпов, например, обнаружите, что забыли написать заголовок какой-либо страницы.

С тестированием веб-ресурса прекрасно справятся программы Macromedia **Dreamweaver** и так называемая «система Денвер». Dreamweaver можно скачать на сайте компании www.macromedia.com, а «Денвер» - по адресу <http://www.dklab.ru>.

После установки обеих программ, скопируйте файлы сайта в папку программы «система Денвер». (по умолчанию это **C:\WebServers\home\папка сайта\www.**). Запустите программу. Откройте Dreamweaver и перейдите к менеджеру сайтов (**Site - Manage Sites**). Щелкните на кнопке **New**, во втором по счету диалоговом окне поставьте флажок напротив пункта «**Yes, I want to use server technology**». Нажмите **Next** и отметьте «**Edit and test locally**». Укажите адрес размещения сайта на компьютере, в данном случае, **C:\WebServers\home\папка сайта\www.**, и URL-адрес сайта (**http://localhost/имя_папки_сайта**). В последнем диалоговом окне установите переключатель **No**. Для просмотра веб-страниц на локальном сервере, в веб-браузере каждый раз придется набирать следующий адрес: http://имя_папки_сайта/имя_веб-страницы.

8. Отсутствие регулярного обновления и архива информации.

Что тут скажешь, если сайт регулярно не обновляется, очень скоро он потеряет постоянных посетителей. Если говорить о старых материалах (статьях, заметках, новостях), стоит отвести для них специальную страницу. Никогда не знаешь наверняка, что заинтересует пользователя, а вдруг ему понадобится какая-либо статья двухгодичной давности.

Заключение.

Всегда лучше учиться на чужих ошибках, и постарайтесь во время верстки сайта не повторять рассмотренные здесь ошибки.

Карта сайта как главный ориентир для пользователя

Непродуманная должным образом навигация веб-сайта, громада всевозможных ссылок и баннеров, беспорядочно сваленных в одну колонку, отсутствие поиска. Подобные «прелести» способны отпугнуть кого угодно. А в условиях жесточайшей конкуренции между сайтами огрехи верстки и дизайна непростительны. Поэтому, пока не поздно, меняем ситуацию.

Навигация.

Для разработки комфортной и удобной системы навигации воспользуйтесь следующими советами:

1. Объединяйте ссылки похожей тематики в блоки.
2. Баннеры и рекламные материалы размещайте в специально отведенном для этих

целей месте на странице.

3. Если ссылок и страниц много, оптимальным вариантом будет создание, с помощью скриптов, выпадающего меню

Карта сайта.

Для карты сайта понадобится создать отдельную страницу. Разберемся, что собой представляет карта сайта.

Карта сайта помогает систематизировать разделы веб-ресурса, страницы и ссылки, содержащиеся в них. Как правило, карта сайта имеет древовидную структуру, напоминающую Проводник операционной системы Windows. То есть создается список разделов веб-ресурса, нажав на которые пользователь получает доступ ко всей информации, в данный момент, находящийся на сайте.

Затем на главную и промежуточные страницы ресурса помещается ссылка на карту сайта в виде текста или графического изображения.

Значение карты сайта трудно переоценить. Кроме комфортного пребывания пользователя на сайте, что немаловажно для роста популярности ресурса, карта сайта также помогает в продвижении ресурса в сети Интернет.

Дело в том, что на одной странице (карте) размещается довольно внушительное количество внутренних ссылок. Внутренние ссылки зачастую представляют для поисковых систем большую ценность, чем внешние. Таким образом, происходит увеличение, и притом существенное, уровня релевантности сайта в глазах поисковых машин.

В связи с этим, карта сайта должна быть обязательным компонентом любого веб-ресурса. Важно не просто сформировать карту сайта, но и грамотно оптимизировать ее, учитывая требования поисковых систем. Внутренние ссылки должны содержать как можно больше ключевых слов, чтобы дать возможность обнаружить новые страницы сайта, которые затем будут включены в процесс индексации. Не надо и говорить, каким образом этот факт отразится на положении ресурса в рейтинге поисковых машин.

Оптимизировать карту сайта можно как вручную, так и с помощью специализированных сервисов. Например, **Google Sitemap** (<http://www.sitemapspal.com>).

Верстка с помощью слоев

Раньше фреймы и таблицы считались основными способами верстки. Но время не стоит на месте, и с выходом нового поколения веб-браузеров, поддерживающих современные технологии, ситуация изменилась. Использование слоев, наподобие тех, что можно встретить в программе Adobe Photoshop, для верстки документов предоставило веб-мастерам поистине неограниченные возможности. Но, если от фреймов и так собирались отказаться ввиду их громоздкости и слишком грубому виду, то между приверженцами табличной верстки и поклонниками слоев разгорелись нешуточные страсти. Дело в том, что у таблиц хоть и существуют недостатки, но из-за простоты освоения этот метод верстки пользуется неизменной популярностью у начинающих веб-мастеров, да и некоторые профессионалы нет-нет, да и применяют табличную верстку на сайте. Поэтому таблицы и слои стали считаться конкурентами. Сейчас, когда ажиотаж вокруг слоев немного поутих, самое время узнать достойный ли у табличной верстки соперник.

Слои являются элементами HTML-кода, которые можно внедрять в веб-страницу путем размещения одного на другом с точностью до пикселя. Изменения в параметры слоя вносятся с использованием языка **JavaScript** или **VBScript**, благодаря чему становится возможным применение на страницах сайта различных эффектов: плавающие окна, вращающиеся надписи, выпадающее меню и другие.

Слой задается контейнером `<Layer>`, а для определения позиции слоя требуются атрибуты **top** (отступ от верхнего края экрана), **left** (отступ от левого края экрана), **z-index** (место расположения отдельного слоя, заданного декартовыми координатами, причем слой с самым большим значением помещается сверху).

Достоинства слоев:

- Метод позиционирования слоев относительно друг друга позволяет создавать интересные эффекты, способствующие «оживлению» веб-страницы.
- Быстрая обработка слоев веб-браузерами.
- Поддержка декартовых координат позволяет указать положение слоев на странице с точностью до пиксела.
- Настройка свойств слоя посредством CSS.

К недостаткам слоев относятся:

- Глубинное знание языков и технологий веб-программирования – JavaScript, VBScript, CSS.
- Различия в отображении сайта, при верстке которого применялись слои, веб-браузерами.
- Проблемы при редактировании страниц со слоями в графических html-редакторах.

Верстка с помощью слоев технически напоминает позиционирование элементов, причем стандартный в таких случаях тег `<Layer>` лучше заменить на `<Div>`, тогда проблем с браузерами поубавится.

Как видите, слои практически не ограничивают фантазию дизайнера, но вместе с тем сложны в исполнении. К тому же единый стандарт восприятия веб-браузерами в них отсутствует. Поэтому никто не гарантирует, что сверстаный таким образом сайт, будет идентично выглядеть и в Internet Explorer, и в Opera. Вот почему до сих пор не отказываются от использования таблиц. Однако, стоит заметить, что в плане верстки у слоев большое будущее, если устроятся разночтения веб-браузеров.

Преимущества верстки с помощью таблиц

Чтобы ни говорили о таблицах: и пользоваться ими неудобно, и качественный сайт с их помощью не сделаешь и вообще, есть более современные технологии, а веб-мастера по-прежнему используют таблицы, и похоже в ближайшее время и не собираются отказываться от этого способа верстки веб-сайтов. Какими преимуществами обладают таблицы и так ли уж существенны их недостатки?

В языке HTML понятие «таблица» имеет два значения. Первое звучит так: таблица – это специальная форма, предназначенная для размещения информации. Второе значение представляет таблицу средством разметки веб-страниц. Нас интересует номер два.

Как и все элементы, таблицы помещаются в веб-страницу посредством HTML-тегов и атрибутов. Причем важно знать, что теги создают каркас таблицы, а атрибуты указывают ее параметры (длину, ширину, толщину линий, цвет заливки).

Начинающие веб-мастера предпочитают таблицы другим способам верстки главным образом

из-за простых методов размещения элементов сайта на веб-странице. Если использование слоев предполагает глубокие знания языков JavaScript или VBScript, то с , таблицами дело обстоит иначе: достаточно знать основы HTML и заранее составить примерную схему макета будущего сайта.

Перечислим основные особенности табличной верстки, а уже на основании полученной информации решим, насколько эффективен подобный метод. Итак, таблицы позволяют осуществлять:

- Выравнивание элементов относительно каркаса таблицы.
- Формирование «резинового» макета (ширина сайта автоматически изменяется в соответствии с разрешением экрана ПК пользователя и параметрами веб-браузера).
- Уменьшение объема веб-страницы путем разрезания изображений на несколько частей. Затем кусочки изображения помещаются в таблицу, где происходит их «склеивание».
- Создание сайта, обладающего свойствами и параметрами, понятными большинству веб-браузеров.

Безусловно, в табличной верстке существуют недостатки (долгая загрузка сайта, слишком объемный код, плохая индексация). Но обойти их довольно легко, например размеры таблицы стоит указывать в процентах, а не в пикселях, использовать нулевую толщину бордюра и всевозможные распорки. А что касается индексации поисковых систем – придется уделить больше времени мета-тегам.

Работа с тегами

Теги: теория

Кроме текстовой и графической информации каждая веб-страница содержит невидимый для пользователя HTML-код или, по-другому, набор тегов. Тегами называются специальные команды языка HTML, которые сообщают веб-браузеру информацию об отдельных элементах страницы (текст, рисунки, таблицы, линии). Веб-браузер в свою очередь использует полученные сведения для отображения элементов страницы в заданном порядке и в соответствии с указанными свойствами.

Тег обозначается одним или несколькими английскими словами, которые объясняют его назначение. Записываются теги следующим образом:

`<Тег>... текст </Тег>` (упрощенный вариант), где `<Тег>` является «открывающим» тегом, а `</Тег>` - «закрывающим». Закрывающие и открывающие теги также называют контейнерами.

Или `<Тег атрибут1="значение" атрибут2="значение".. атрибутN="значение">текст</Тег>` (более сложный код с использованием атрибута – набор символов, соответствующий какому-либо английскому слову, дающему понять предназначение атрибута).

Практическая часть.

Если вы хотя бы в общих чертах знакомы с синтаксисом и базовыми элементами языка HTML, проблем при работе с тегами возникнуть не должно. Поэтому, когда основные работы по верстке веб-сайта будут закончены, стоит задуматься о кое-каких деталях, а

заодно на практике закрепить навыки работы с тегами.

1. Мета-теги

С помощью этого вида тегов в HTML-код вводится информация о содержании сайта, его владельце, авторе дизайна. Данные отображаются в разделе «description», являющегося объектом пристального внимания поисковых роботов. Да, не стоит удивляться, именно на данные мета-тегов в первую очередь ориентируются роботы и на основании полученной информации отображают сайт в результатах поиска. Поэтому так важно добавлять в исходный код не только описание ресурса, но и ключевые слова: `<meta name="keywords" content="дизайн, веб-дизайн, программирование" />`

Также возможны и более сложные манипуляции. Например, для обеспечения регулярного просмотра поисковыми роботами вашей веб-страницы необходимо ввести следующий текст: `<meta name="document-state" content="dynamic" />`

В таблице приведены основные мета-теги и их значения. Внимательно изучите следующие особенности мета-тегов, поскольку в дальнейшем полученные знания можно применить для оптимизации сайта под поисковые системы.

Тег	Описание
Owner	владелец ресурса
Author	автор и разработчик ресурса
Description	краткое описание сайта
Copyright	авторские права
Generator	программное обеспечение, с помощью которого создавался сайт
Keywords	ключевые слова

2. Добавление страницы в избранное

Чтобы пользователь одним нажатием на кнопку смог добавить понравившийся веб-ресурс в папку «Избранное» веб-браузера, примените следующий код:

```
<a href="javascript:window.external.AddFavorite ('http://адрес_страницы', 'описание_страницы')">Добавить в избранное</A>
```

4. Установка домашней страницы

Обычно домашними страницами назначают поисковые системы, или каталоги, чтобы в случае необходимости быстро найти информацию. Однако ссылка для установки домашней страницы может присутствовать и на обычном сайте. Ниже приведен необходимый html-код.

```
<head>
```

```
<title>JavaScript. Установление страницы стартовой</title>
```

```
</head>
```

```
<body>
```

```
<a
```

```
href="#"
```

```
onClick="this.style.behavior='url(#default#homepage)';this.setHomePage('http://');  
return false;">Сделать стартовой</a>
```

```
</body>
```

```
</html>
```

4. Перенаправление страниц

С перенаправлением пользователя по истечению определенного количества времени на другую страницу, вы наверняка не раз встречались при загрузке контента. Для использования на сайте подобного метода введите код: `<meta http-equiv="refresh" content="время в секундах; url=адрес">`.

5. Значок сайта

Обратите внимание: при загрузке некоторых веб-ресурсов в адресной строке отображается значок сайта. Делается это с целью формирования в сознании пользователя индивидуального стиля ресурса.

Код значка выглядит так: `<link rel="shortcut icon" href="favicon.ico">`
Теперь ваш сайт полностью готов.

ЛИСТ
изменений рабочей учебной программы по дисциплине
ОП.12 ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ. WEB
ПРОГРАММИРОВАНИЕ

Дополнения и изменения, вносимые в рабочую программу дисциплины

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 №2.4.5 и №5 Перечня основной и дополнительной учебной литературы	Обновлен список рекомендуемой литературы

Составитель: преподаватель

Т.П. Кривошеенко

Утвержден на заседании предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах протокол № 10 от «27» мая 2020 г.

Председатель предметной (цикловой) комиссии
профессиональных дисциплин специальности
Программирование

в

компьютерных

системах

Л.А. Благова

подпись

«27» мая 2020 г.

Зам. директора по УР филиала

Т. А. Резуненко

«27» мая 2020 г.

Заведующая сектором библиотеки

Л. Г. Соколова

«27» мая 2020 г.

Инженер-электроник (программно-информационное обеспечение образовательной программы)

А. В. Сметанин

«27» мая 2020 г.

Рецензия

на рабочую программу учебной дисциплины
ОП.12 «Интернет программирование. Web-программирование»
для специальности 09.02.03 Программирование в компьютерных системах

Рабочая программа учебной дисциплины ОП.12 «Интернет программирование. Web-программирование» разработана на основе Федерального государственного образовательного стандарта (далее - ФГОС) по специальности среднего профессионального образования 09.02.03 Программирование в компьютерных системах, утвержденного приказом Минобрнауки Российской Федерации от 28.07.2014 № 804 зарегистрирован в Минюсте России 21.08.2014 (рег. № 33733).

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 динамической 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;

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

Содержание рабочей программы учебной дисциплины полностью соответствует требованиям Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах. Изучение данной дисциплины способствует эффективной и качественной подготовке молодых специалистов в области интернет программирования.

Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники, изданные в последнее время (не позднее 5 лет).

Разработанная программа учебной дисциплины ОП.12 «Интернет программирование. Web-программирование» может быть рекомендована для использования в учебном процессе при подготовки по специальности 09.02.03 Программирование в компьютерных системах.

Директор ООО «Современные
информационные технологии»

А.В.Сметанин

Рецензия

на рабочую программу учебной дисциплины

ОП.12 «Интернет программирование. Web-программирование»

для специальности 09.02.03 Программирование в компьютерных системах

Рабочая программа учебной дисциплины ОП.12 «Интернет программирование. Web-программирование» разработана на основе Федерального государственного образовательного стандарта (далее - ФГОС) по специальности среднего профессионального образования 09.02.03 Программирование в компьютерных системах, утвержденного приказом Минобрнауки Российской Федерации от 28.07.2014 № 804 зарегистрирован в Минюсте России 21.08.2014 (рег. № 33733).

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 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. 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: 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 09.02.03 Программирование в компьютерных системах. Анализ раздела «Условие реализации программы учебной дисциплины», позволяет сделать вывод, что образовательное учреждение располагает материально-технической базой, отвечающей современным требованиям подготовки специалистов, обеспечивает проведение всех видов практических занятий, междисциплинарной подготовки, 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 «Интернет программирование. Web-программирование» может быть рекомендована для использования в учебном процессе при подготовки по специальности 09.02.03 Программирование в компьютерных системах.

Рецензент

Программист ГБУЗ

"Геленджикский психоневрологический диспансер"
министерства здравоохранения Краснодарского края

Е.В.Мельников