

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет информатики и информационных технологий

Рабочая программа дисциплины

«WEB технологии»

Кафедра Информатики и Информационных технологий

Образовательная программа

информационные системы и технологии 09.03.02

Профиль подготовки: Информационные системы и технологии

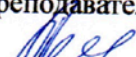
Уровень высшего образования: бакалавриат

Форма обучения: очная

Статус дисциплины: вариативная по выбору

Махачкала 2017

Рабочая программа дисциплины «WEB технологии» составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки **09.03.02 – Информационные системы и технологии**, уровень подготовки: бакалавриат, утвержденного приказом Минобрнауки РФ от 12 марта 2015 г. №219.

Составитель: Ахмедова Написат Мурадовна, старший преподаватель кафедры информатики и информационных технологий. 


Рабочая программа одобрена на заседании кафедры информационных систем и технологий

17.08.2017 г., протокол № 8.

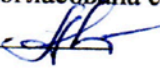
Заведующий кафедрой  Ахмедов С.А.

Рабочая программа одобрена на заседании Методической комиссии факультета информатики и информационных технологий

31.08.2017 г., протокол № 7.

Председатель методсовета факультета 

Рабочая программа согласована с учебно-методическим управлением

31.08.2017 г., 

Аннотация рабочей программы дисциплины

Дисциплина Web-технологии входит в вариативную часть по выбору образовательной программы бакалавриата по направлению 09.03.02 Информационные системы и технологии.

Дисциплина реализуется на факультете ИиИТ кафедрой ИиИТ.

Содержание дисциплины охватывает круг вопросов связанных с базовыми концепциями и приемами web -программирования, использованием современных web-технологий, в частности: CGI, Ajax. Использование современных языков для создания web-приложений, таких как: HTML, CSS, JavaScript, DHTML, PHP, Perl. Создание web-сервисов, сайтов, порталов с использованием этих технологий.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных – ОПК-5, профессиональных – ПК-34.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, семинарские занятия, лабораторные работы и самостоятельную работу.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме контрольных работ и промежуточный контроль в форме зачета.

Объем дисциплины 3 зачетные единицы, в том числе в академических часах по видам учебных занятий:

Се- местр	Учебные занятия						СРС, в том числе экза- мен	Форма промежу- точной аттеста- ции (зачет, диф- ференцированны й зачет, экзамен)
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Все- го	из них						
		Лек- ции	Лаборатор- ные заня- тия	Практи- ческие занятия	КСР	консуль- тации		
6	108	18	18	18	6		48	зачет

1. Цели освоения дисциплины

Целью освоения дисциплины WEB технологии являются: знакомство с базовыми

концепциями и приемами web -программирования, научить использовать современные web-технологии, в частности: CGI, Ajax. Научить использовать современные языки для создания web-приложений, такие как: HTML, CSS, JavaScript, DHTML, PHP, Perl. Научить создавать web-сервисы, сайты, порталы с использованием этих технологий.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина WEB технологии входит в вариативную часть по выбору образовательной программы бакалавриата по направлению 09.03.02 Информационные системы и технологии.

Для эффективного освоения дисциплины требуются знания по информатике, основам программирования, а также основам построения информационных систем.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения) .

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
ОПК-5	способностью использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению	знат ь: <ul style="list-style-type: none"> • принципы гипертекстовой разметки; • структуру HTML-документа; • основные теги языка HTML; • основные принципы каскадных таблиц стилей; • язык JavaScript и его особенности; • основные особенности языка PHP; заголовки HTTP;
ПК-34	способностью к инсталляции, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию	умет ь: <ul style="list-style-type: none"> • составлять HTML-документ средствами языка HTML; • применять каскадные таб-

		лица стилей; <ul style="list-style-type: none"> • применять JavaScript для контроля данных, введенных в форму; • обрабатывать формы; • работать с cookies; Имеет опыт : <ul style="list-style-type: none"> • создания web-приложений; • создания web-сервисов, сайтов, порталов.
--	--	---

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 3 зачетные единицы, 108 академических часов.

4.2. Структура дисциплины.

Разделы дисциплины	Трудоемкость освоения раздела дисциплины, час						
	семестр	номер модуля, в который включен раздел	Лекции	Практические занятия	Лабораторные работы	КСР	Самостоятельная работа
1. История и основные тенденции развития Web – технологий. Принципы гипертекстовой разметки и каскадные таблицы стилей CSS.	6	1	6	6	6	2	16
2. Язык JavaScript и динамический HTML.	6	2	6	6	6	2	16
3. Серверное программирование.	6	3	6	6	6	2	16
Зачет							
Итого по дисциплине	108		18	18	18	6	48

4.3. Содержание дисциплины, структурированное по темам (разделам).

1. Введение. Принципы гипертекстовой разметки и каскадные таблицы стилей CSS.

1.1. История и основные тенденции развития Web – технологий.

История и основные тенденции развития web – технологий. История гипертекста. Возникновение и развитие службы WWW. История браузеров. Классическая архитектура службы WWW и ее составляющие.

1.2. Гипертекстовая разметка, структура HTML-документа.

Принципы гипертекстовой разметки. Язык HTML. Теговая модель: контейнеры, теги, атрибуты, сущности. Группы элементов HTML Гиперссылки.

1.3. Каскадные таблицы стилей CSS.

Каскадные таблицы стилей CSS. Назначение. Способы применения. Виды CSS-селекторов. Определение и использование селекторов. Наследование и переопределение стилей. Блочные и строковые элементы. Свойства блоков текста. Свойства строковых элементов. Позиционирование с помощью CSS. Методы позиционирования компонентов HTML-страниц: таблицы, фреймы, CSS.

2. Язык JavaScript и динамический HTML.

2.2. Язык JavaScript.

Основные синтаксические конструкции: условный оператор, циклы. Функции. Объекты. Массивы, строки. Регулярные выражения и их использование. Тег `<script>`. Программирование реакции на события. Применение JavaScript для контроля данных, введенных в форму.

2.3. Динамический HTML и Объектная модель документа (DOM).

Программирование окон и фреймов, свойства документа, работа с формами. Доступ к элементам страницы, модификация элементов и их атрибутов, работа со стилями. Обработка событий. Использование объектной модели документа.

3. Серверное программирование.

3.4. Обзор возможностей языка PHP.

Языки программирования PHP. Основные синтаксические конструкции. Регулярные выражения. Особенности языка.

3.5. Регулярные выражения.

Использование регулярных выражений в языках PHP и JavaScript.

3.6. Обработка форм, использование cookies, организация сеансов

работы пользователей.

Генерация HTML-документа, чтение заголовков HTTP и переменных окружения CGI, обработка форм, генерация заголовков HTTP, использование cookies, организация сеансов работы пользователей.

4.4. Программа практических и лабораторных занятий

к разделу «Принципы гипертекстовой разметки и каскадные таблицы стилей CSS»

1. Форматирование символов.

Используемые элементы: **b**, **i**, **u**, **font**, **p**, **br**.

Выполнить задание по образцу, содержащемуся в файле "ex1.html".

2. Вставка графики и обтекание текстом.

Проанализируйте назначение некоторых атрибутов элемента **img**:

SRC=url

ALIGN=**ABSBOTTOM** | **ABSMIDDLE** | **BASELINE** | **BOTTOM** | **LEFT** |
MIDDLE | **RIGHT** | **TEXTTOP** | **TOP**

ALT=text

WIDTH=n

HEIGHT=n

TITLE=text

Выполнить задание по образцу, содержащемуся в файле "ex2.html".

3. Таблицы. Атрибут **width**.

Создайте таблицу, состоящую из одной строки и двух ячеек в ней. В элементе **table** укажите **border=1**. Поэкспериментируйте со значениями атрибута **width** в элементах **table** и **td**. В частности, проверьте следующие значения (для сравнения можно создать несколько таблиц в одном документе):

```
table td1 td2
```

```
400 200 200
```

```
400 200 100
```

```
400 100 100
```

```
200 150 150
```

150 150

Определите ширину таблицы.

100% 50% 50%

100% 20% 80%

100% 20% 20%

80% 20%

80% 80%

100% 100 100

100% 100

100% 100

100% 50% 100

100% 100 50%

100% 100 10%

400 10%

400 100 10%

10% 100

Определите ширину таблицы.

Сформулируйте правила определения ширины ячеек в таблице с двумя ячейками.

4. Объединение ячеек и вложенные таблицы. Используемые элементы: **table**, **tr**, **td**. Используемые атрибуты:

`width`, `cellpadding`, `cellspacing`, `border`, `gcolor`, `colspan`, `rowspan`.

Выполнить задание по образцу, содержащемуся в файле "ex4.html".

5. Применение таблиц для позиционирования компонентов страницы. Используемые элементы (атрибуты):

`table(width)`, `tr`, `i`, `font(size)`, `td(width, colspan, align)`, `br`, `center`, `div(align)`.

Указание: для задания выравнивания по ширине и отступа первой строки в последнем абзаце использовать

CSS: `<p style="text-align:justify; text-indent:50;">`

Выполнить задание по образцу, содержащемуся в файле "ex5.html".

6. Навигационные компоненты страницы.

Образец: "ex6.ml".

Определите стили для отображения следующих компонентов: ячейка таблицы с пунктом главного меню, общий стиль ссылки, ссылка главного меню (с эффектом наведения и выделением текущего раздела), ссылка меню второго уровня (с выделением текущего подраздела), заголовок текста, текст страницы.

7. Абсолютное и относительное позиционирование.

Образец: "ex2.html".

Указание: Каждый рисунок встроен в страницу с помощью элемента div:

```
<div style="">  
  
</div>
```

Нужно определить соответствующий стиль отображения для каждого из восьми таких блоков, учитывая, что

1) шесть блоков слева размещены по трем ячейкам таблицы, первые два из них позиционированы по умолчанию, а остальные имеют относительное позиционирование;

2) два блока справа имеют абсолютное позиционирование, их можно вставить в страницу в произвольном месте.

8. Верстка с помощью CSS.

Выполнить задание по образцу, содержащемуся в файле "ex3.html".
Содержимое элемента body:

```
<div class="title">Повести покойного<br>Ивана Петровича Бел-  
кина</div>  
<div class="epigraph">  
<div class="person">Г-жа Простакова</div>  
<div class="cue">То, мой батюшка, он еще сызмала к историям охот-  
ник.</div>  
<div class="person">Скотинин</div>  
<div class="cue">Митрофан по мне.</div> <div  
class="source">Недоросль</div></div> <div  
class="subtitle">ОТ ИЗДАТЕЛЯ</div> <div  
class="text">
```

Взявшись хлопотать об издании Повестей И. П. Белкина, предлагаемых ныне публике, мы желали к оным присовокупить, хотя краткое жизнеописание покойного автора и тем отчасти удовлетворить справедливому любопытству любителей отечественной словесности. Для сего обратились, было, мы к Марье Алексеевне Трафилиной, ближайшей родственнице и наследнице Ивана Петровича Белкина; но, к сожалению, ей невозможно было нам доставить никакого о нем известия, ибо покойник вовсе не был ей знаком. Она советовала нам отнестись по сему предмету к

одному почтенному мужу, бывшему другом Ивану Петровичу. Мы последовали сему совету, и на письмо наше получили нижеследующий желаемый ответ. Помещаем его безо всяких перемен и примечаний, как драгоценный памятник благородного образа мнений и трогательного дружества, а вместе с тем, как и весьма достаточное биографическое известие.

</div>

к разделу «Язык JavaScript и динамический HTML»

Указание к выполнению работы

Все задания этой работы необходимо оформить в виде отдельных js-файлов, выполняющихся с помощью сервера сценариев Windows.

Исходные данные можно присваивать непосредственно в начале сценария, а можно также использовать аргументы командной строки, доступные через свойство WScript.Arguments:

```
args = WScript.Arguments;  
arg0 = args(0);  
arg1 = args(1); // и т.д.  
//args.length - количество аргументов
```

Во втором случае сценарий удобнее запускать из интерпретатора команд (cmd).

Для вывода результатов нужно использовать метод WScript.Echo(str). 1. Явное и неявное преобразование типов.

Определите тип результата в каждой строке и сформулируйте правила неявного преобразования типов:

```
a=10; b="10"; res=a+b;  
a="10"; b=10; res=a+b;  
a="10"; b="10"; res=a+b;  
a=10; b=10; res=a+b;  
a="10"; b=true; res=a+b;  
a=10; b=true; res=a+b.
```

Проанализируйте действие методов parseInt и parseFloat:

```
a="10"; b=parseInt(a); res=1+b;  
a=10; b=parseInt(a); res=1+b; a="10abc";  
b=parseInt(a); res=1+b; a="10.15";  
b=parseInt(a); res=1+b; a="abc";
```

```
b=parseInt(a); res=1+b; a="10.15";
```

```
b=parseFloat(a); res=1+b.
```

Указание: Если эти методы не могут преобразовать число в строку, то они возвращают специальное значение **NaN** (Not a Number). Проверить это значение можно методом `isNaN(number)` булевского типа. Можно также использовать метод `isFinite(number)`, возвращающий `true`, если число не имеет значение **NaN** и не является положительной или отрицательной бесконечностью.

Преобразовать число в строку можно так:
`a=10.15; str=a.toString();`

2. Работа со строками: скобки одного вида.

Дана строка, содержащая латинские буквы и круглые скобки. Если скобки расставлены правильно (то есть каждой открывающей соответствует одна закрывающая), то вывести число 0. В противном случае вывести или номер позиции, в которой расположена первая ошибочная закрывающая скобка, или, если закрывающих скобок не хватает, число -1.

3. Работа со строками: скобки трех видов.

Дана строка, содержащая латинские буквы и скобки трех видов: "()", "[]", "{}". Если скобки расставлены правильно (то есть каждой открывающей соответствует закрывающая скобка того же вида), то вывести число 0. В противном случае вывести или номер позиции, в которой расположена первая ошибочная скобка, или, если закрывающих скобок не хватает, число 1.

4. Строки и массивы: посимвольная обработка строк.

Известна функция, осуществляющая декодирование некоторой строки: **function decode(str) {**

```
var arr = str.split(","); var res = "";
```

```
for(var i=0; i<arr.length; i++)
```

```
res += String.fromCharCode(arr[i]); return res; };
```

Проанализируйте ее работу, определите формат закодированной строки и напишите соответствующую функцию для кодирования.

Указание: Метод `split` предназначен для разбиения исходной строки на части. Вот его описание:

```
arrObj = stringObj.split([separator[, limit]]);
```

stringObj исходная строка;

arrObj массив подстрок;

separator разделитель подстрок в исходной строке, если его опустить, то возвратится одноэлементный массив, содержащий все строку;

Limit определяет Наибольшее количество элементов в результирующем массиве. Если предел достигнут, то последний элемент массива содержит весь остаток строки независимо от наличия в нем разделителей.

5. Строки и массивы: анализ выражения.

Даны две строки, задаваемые следующей грамматикой:

String ::= Expression | (Expression '&' String)

Expression ::= Name '=' Value

Name ::= (Letter)+

Value ::= '0' | ([1-9](Digit)*)

Letter ::= [a-z]

Digit ::= [0-9]

Пример (*):

```
str1 = "a=10&bvf=654&abc=0";
```

```
str2 = "c=15&a=24&abc=10".
```

1) Вычислить сумму всех значений (Value).

2) Сформировать строку - список всех имен (**Name**) в алфавитном порядке, разделенных запятой (каждое имя должно встречаться столько раз, сколько оно присутствует в исходных строках).

3) Сформировать строку - список всех имен без повторений вместе со значениями (если имена повторяются, то вычислять сумму значений).

В примере (*) должно получиться: res

```
== "a=34,abc=10,bvf=654,c=15"
```

Указание: удобно воспользоваться методами `split`, `join`, `push`.

6. Работа с массивами: использование методов `slice` и `splice`.

Назовем центральной окрестностью радиуса m часть массива (срез), заключенную между элементами с номерами $(N-1)/2-m$ и $(N-1)/2+m$ (где N - количество элементов массива), включая сами эти элементы. Даны два массива с нечетным количеством элементов и целые числа $m1$ и $m2$. Поменять местами центральные окрестности радиусов $m1$ и $m2$ заданных массивов.

к разделу «Серверное программирование»

Указание к выполнению работы

Все задания этой работы нужно выполнять в HTML-документе, содержащем тестовые элементы. Решения оформлять в виде функций в элементе SCRIPT заголовка документа и вызывать их из обработчиков событий onload или onclick тела документа. Ответы выводить в специально оформленный блок страницы.

Способы использования регулярных выражений.

1) Проверка на соответ-

```
ствие. var s =  
"abcxxxabc"; var re =  
/x+;/ if(s.match(re)){
```

```
//Строка s соответствует шаблону re.  
}
```

2) Выделение подстрок при однократном поиске. var s = "a1122233b12333";

```
var re = /(1+)(2+)(3+)/;  
var arr = s.match(re);
```

```
// Проверьте содержимое массива arr и значения свойств  
// arr.input, arr.index, arr.lastIndex.
```

3) Выделение подстрок при глобальном поиске.

```
var s = "a11b11def1111gh";
```

```
var re = /1+/g;
```

```
var arr = s.match(re);
```

```
// Проверьте содержимое массива arr и значения свойств  
// arr.input, arr.index, arr.lastIndex.
```

```
// Сравните с предыдущим пунктом
```

4) Замена части строки.

```
var s = "a11b22c333"; var
```

```
re = ^\d+/;
```

```
var s1 = s.replace(re, "x");
```

```
// Проверьте, изменилась ли строка s.
```

```
// Проведите глобальную замену.
```

1. Составьте следующие регулярные выражения:

- 1) полный телефонный номер сети Megafon (напр. "(+7928)999-99-99");
модифицируйте регулярное выражение, чтобы можно было набирать и (+7...) и (8...).
- 2) номенклатура электрического кабеля в формате: "марка"- "напряжение"- "число жил"x"сечение",
где "марка" - может принимать значения NYM, AVVG, PPV;
"напряжение" - число 1 или 2; "число жил" - число от 2 до 5;
"сечение" - любое целое число.
- 3) фамилия с одним, двумя или тремя инициалами (англ.) с выделением каждого элемента имени по отдельности (напр. "J.F.Kennedy");
- 4) выделение ссылки (href) из элемента a (напр. "link").

2. Коллекция **all**.

Дан документ в формате **html**. Найти в нем количество элементов каждого типа.

3. Коллекция **children**.

Дан документ в формате **html**. Найти глубину и ширину дерева иерархии объектов (глубина - это длина самой длинной ветви, ширина - наибольшее количество узлов на одном уровне). Вывести список элементов, составляющих самую длинную ветвь и номер самого широкого уровня.

4. Позиционирование блока.

Используя свойства **offsetLeft**, **offsetTop** и **offsetParent** некоторого блочного элемента организовать случайное изменение его положения при каждом щелчке мышью.

5. Динамическое изменение содержимого - 1.

Выведите на пустую страницу следующие градиенты: от белого до черного цвета, от зеленого до красного цвета, от черного до синего цвета.

6. Динамическое изменение содержимого Сформируйте и выведите на страницу блочные элементы для игры в

"пятнашки" (пример в файле "15.png").

7. Проверка корректности данных, введенных в форму. Образец: "ex2.html". Эле-

менты форм обычно именуются с помощью атрибута name. Доступ к ним осуществляется через коллекцию forms документа.

Перед отправкой формы обычно проверяют ее содержимое. Для этого нужно использовать обработчик события onsubmit формы. Если обработчик возвращает **false**, то отправка данных формы отменяется.

3. Координаты указателя мыши.

Пользуясь документом ex3.html установите смысл следующих свойств объекта event: x, y, clientX, clientY, offsetX, offsetY, screenX, screenY.

4. Напишите игру "Пятнашки". Перемещение блоков удобно организовать с помощью обработчиков событий onmouseup и onmousedown.

5. Образовательные технологии

Рекомендуемые образовательные технологии: лекции, лабораторные занятия, самостоятельная работа студентов.

В соответствии с требованиями ФГОС ВО по направлению подготовки реализации компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (компьютерных симуляций, разбор конкретных ситуаций) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Удельный вес занятий, проводимых в интерактивных формах, определяется главной целью (миссией) программы, особенностью контингента обучающихся и содержанием конкретных дисциплин, и в целом в учебном процессе они должны составлять не менее 30% аудиторных занятий (определяется требованиями ФГОС с учетом специфики ООП). Занятия лекционного типа для соответствующих групп студентов не могут составлять более 60% аудиторных занятий (определяется соответствующим ФГОС)).

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Для самостоятельной подготовки студентов рекомендуется использовать ресурс дистанционного обучения <http://moodle.dgu.ru/>, курс: WEB технологии. В данной системе представлены материалы лекций в электронном виде, а также подготовлены электронные тесты, позволяющие оперативно оценить уровень знаний студента. Методические указания к выполнению лабораторных работ также можно найти в данном разделе системы дистанционного обучения.

Порядок и последовательность самостоятельного изучения материалов определены в системе ДО moodle.dgu.ru, курс: WEB технологии.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

№	Названия разделов	Са-мост. Рабо-та, час.	Виды и содержание самостоятельной работы
1	2	3	4
	Модуль 1		
1	Язык JavaScript.	8	<ul style="list-style-type: none"> • проработка учебного материала • работа с тестами и вопросами для самопроверки
2	Гипертекстовая разметка, структура HTML-документа.	8	<ul style="list-style-type: none"> • проработка учебного материала • работа с тестами и вопросами для самопроверки
3	Каскадные таблицы стилей CSS.	8	<ul style="list-style-type: none"> • проработка учебного материала • работа с тестами и вопросами для самопроверки
	Модуль 2		
4	Динамический HTML и Объектная модель документа (DOM).	12	<ul style="list-style-type: none"> • проработка учебного материала • работа с тестами и вопросами для самопроверки
5	Обзор возможностей языка PHP.	12	<ul style="list-style-type: none"> • проработка учебного материала • работа с тестами и вопросами для самопроверки
	Всего часов	48	

Форма контроля и критерий оценок

В соответствии с учебным планом предусмотрен зачет в 6 семестре.

Формы контроля: текущий контроль, промежуточный контроль по модулю, итоговый контроль по дисциплине предполагают следующее распределение баллов.

Текущий контроль

Посещаемость занятий 5 баллов

Выполнение 1 лабораторной работы 10 баллов

Промежуточный контроль

По завершении модуля проводить устный опрос 60 баллов

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

Компетенция	Знания, умения, навыки	Процедура освоения
ОПК-5	<p>знает б.:</p> <ul style="list-style-type: none">• принципы гипертекстовой разметки;• структуру HTML-документа;• основные теги языка HTML;• основные принципы каскадных таблиц стилей;• язык JavaScript и его особенности;• основные особенности языка PHP;• заголовки HTTP;	Устный опрос
ПК-34	<p>умет б.:</p> <ul style="list-style-type: none">• составлять HTML-документ средствами языка HTML;• применять каскадные таблицы стилей;• применять JavaScript для контроля данных, введенных в форму;• обрабатывать формы;• работать с cookies; <p>имет б опыт :</p> <ul style="list-style-type: none">• создания web-приложений;• создания web-сервисов, сайтов, порталов.	Лабораторные работы

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания.

ОПК-5

Схема оценки уровня формирования компетенции «способность к установке, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию»

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Пороговый	Способность к установке, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию	Допускаются ошибки при анализе информации. Неспособен применить на практике изученные технологии. Способен аргументированно представить результат.	Не допускает ошибок при анализе результатов. Способен аргументировать полученные результаты. Способен самостоятельно изучить новые технологии и понимает как их применить на практике.	Проводит быстрый и качественный поиск исходных данных, способен самостоятельно реализовать приложение. В коллективе способен отстаивать свою точку зрения путем аргументации. Обладает способностью разделять задачи между коллегами и контролировать ход выполнения.

ПК-34

Схема оценки уровня формирования компетенции «способность к установке, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию»

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Пороговый	Способность к установке, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию.	Знание основных угроз и средств борьбы с ними. Допускаются ошибки при практическом применении средств WEB.	Знание основных угроз и средств борьбы с ними. Умение управлять ЗИ средствами организационных решений. Допускаются ошибки при практическом применении средств WEB.	Умение применять современные средства Web и оценивать их надежность. Знание основных технологий. Умение создавать приложения современными web-технологиями.

Если хотя бы одна из компетенций не сформирована, то положительная оценка по дисциплине быть не может.

7.3. Типовые контрольные задания

ПЕРЕЧЕНЬ КОНТРОЛЬНЫХ ВОПРОСОВ ДЛЯ ПОДГОТОВКИ К ИТОГОВОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

1. Язык HTML. Теговая модель: контейнеры, теги, атрибуты, сущности. Гиперссылки.
2. Язык программирования PHP. Интеграция с базами данных.
3. Структура HTML-документа. Элементы заголовка. Элементы тела документа.
4. Язык программирования PHP. Регулярные выражения.
5. Язык HTML. Основные теги. Списки. Изображения.
6. Язык программирования PHP. Функции для работы со строками.
7. Использование таблиц для верстки HTML-страниц. Фреймы.
8. Язык программирования PHP. Функции обработки массивов. Математические функции.
9. Формы HTML: основные элементы, свойства и события.
10. Язык программирования PHP. Выражения. Операторы. Основные синтаксические конструкции.
11. Каскадные таблицы стилей CSS. Назначение. Способы применения. Виды CSS-селекторов. Определение и использование селекторов.
12. Применение JavaScript для контроля данных, введенных в форму.
13. Каскадные таблицы стилей CSS. Наследование и переопределение стилей.
14. Язык JavaScript. Объектная модель документа (DOM).
15. Каскадные таблицы стилей CSS. Блочные и строковые элементы. Свойства блоков текста.
16. JavaScript в HTML. Тег <script>. Программирование реакции на события.
17. Каскадные таблицы стилей CSS. Свойства строковых элементов. Позиционирование с помощью CSS.
18. Язык JavaScript. Основные синтаксические конструкции.
19. Язык JavaScript. Применение. Выражения и операции. Операторы.
20. Понятие CGI-скрипта. Типы запросов. Механизмы приема данных скриптом. Механизм генерации отклика скриптом. Переменные CGI – окружения.

ПЕРЕЧЕНЬ ТЕМ РЕФЕРАТОВ

- 1) Развитие Web-технологий
- 2) Программирование на языке Java Script
- 3) Руководство по HTML
- 4) Примеры использования Java и CGI скриптов
- 5) Создание Web-сайта
- 6) Создания интерактивных WEB - страниц для получения информации в сети Интернет с использованием форм, заполняемых пользователем

- 7) Язык html: системы программирования html, особенности языка и реализация системы. Назначение, примеры программирования приложений
- 8) Язык разметки гипертекста
- 9) HTML-документы
- 10) Приемы программирования на JavaScript
- 11) Создание web-страниц, введение в стандарты HTML и PHP, элементы технологии WWW
- 12) Разработка сайта с использованием CSS
- 13) Использование JAVA-технологий для разработки графических приложений
- 14) Сети электровычислительных машин и телекоммуникаций
- 15) Применение технологии Macromedia Flash для разработки сайта
- 16) Создание веб-сайта по технологии flash
- 17) Java — язык программирования
- 18) Технологии программирования Web
- 19) Средства создания Web-приложений
- 20) Базовые стандарты технологий Web
- 21) Основные правила и этапы создания Web-сайтов
- 22) Работа в HTML-редакторе
- 23) Особенности стандарта HTML5
- 24) Web-браузеры

ПРИМЕР ТЕСТОВЫХ ЗАДАНИЙ

1. Как расшифровывается HTML?
 - а) HyperText Markup Language
 - б) HyperThread Mask Language
 - в) HyperThread Markup Language
 - г) HyperText Mask Language
2. Какое расширение должны иметь HTML документы?
 - а) .html или .htm
 - б) .php или .asp
 - в) .txt или .doc
 - г) .doc
3. Какой тег позволяет вставлять картинки в HTML документы?
 - а)
 - б) <pic>
 - в) <picture>
 - г) <image>
4. Выберите ссылку с правильным синтаксисом.

- а) `wisdomweb.ru`
- б) ``
- в) `wisdomweb.ru`
- г) `wisdomweb.ru`

5. С помощью какого тэга можно сделать текст жирным?

- а) ``
- б) `<i>`
- в) `<big>`
- д) `<h2>`

5. Укажите тэг позволяющий создавать заголовки.

- а) `<h2>`
- б) ``
- в) `<small>`
- г) ``

6. Web-страница (документ HTML) представляет собой:

- а) Текстовый файл с расширением htm или html
- б) Текстовый файл с расширением txt или doc
- в) Двоичный файл с расширением com или exe
- г) Графический файл с расширением gif или jpg

7. Для просмотра Web-страниц в Интернете используются программы:

- а) Internet Explorer или Opera (Google Chrome)
- б) MicroSoft Word или Word Pad
- в) MicroSoft Access или MicroSoft Works
- г) HTMLPad или Front Page

8. Тег - это:

- а) Специальная команда, записанная в угловых скобках `< >`
- б) Текст, в котором используются спецсимволы
- в) Указатель на другой файл или объект
- г) Фрагмент программы, включённой в состав Web-страницы

9. Укажите тег позволяющий определить таблицу.

- а) `<table>`
- б) `<tab>`
- в) `<tr>`
- г) `<tabs>`

10. Для вставки изображения в документ HTML используется команда:

- а) ``
- б) `<body background="ris.jpg">`
- в) ``
- г) `<input="ris.jpg">`

11. Гиперссылка задается тегом:

- а) `текст`
- б) ``
- в) ``
- г) `<embed="http://www.da.ru">`

12. Укажите тег позволяющий определить упорядоченный список.

- а) ``
- б) ``
- в) ``
- г) `<list>`

13. Ссылка на адрес электронной почты задается тегом:

- а) `текст`
- б) `kompas@email.ru`
- в) `текст`
- г) `<piter@mailru.com>`

14. Укажите тег позволяющий определить неупорядоченный список.

- а) ``
- б) ``
- в) ``
- г) `<list>`

15. Какие атрибуты использовались при создании данной таблицы?

Ячейка номер 1	Ячейка номер 2
Растянутая ячейка номер 3	

- а) `border='1'` и `colspan='2'`
- б) никакие
- в) `border='1'` и `rowspan='1'`
- г) `border='0'` и `rowspan='2'`

16. Какие тэги использовались при создании текста ниже?

*Здесь вечный отдых для меня начнется.
И здесь стряхну ярмо зловещих звезд
С усталой шеи. — В последний раз,
Глаза, глядите; руки, обнимайте!
Вы, губы, жизни двери, поцелуем
Скрепите договор с корыстной смертью!"*

"Ромео и Джульетта" У. Шекспир

- а) `<pre>` и `<i>`
- б) `<i>`

в) <p>

г) <i> и <p>

17. Какие тэги были использованы при создании формы ниже?

Текстовое

Флажок 1 Флажок 2

Радио кнопка 1 Радио кнопка 2

а) <input type='text'>, <input type='checkbox'> и <input type='radio'>

б) <input type='textfield'>, <input type='checkbox'> и <input type='radiobutton'>

в) <input type='text'> и <input type='check'>

г) <input type='text'>, <input type='check'> и <input type='radio'>

7.4. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий - 3 баллов,
- участие на практических занятиях - 3 баллов,
- выполнение лабораторных заданий – 10 баллов,
- выполнение домашних (аудиторных) контрольных работ - 5 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 3 баллов,
- письменная контрольная работа - 5 баллов,
- тестирование - 10 баллов.

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

Основная литература

1. А.В. Фролов, Г.В. Фролов. Сервер Web своими руками /М: Диалог-МИФИ, 1997
2. Стивен Спейнауэр, Роберт Экштейн. Справочник Вебмастера /СПб: Символ, 2013
3. Ален Вайк, Джейсон Джиллиам. Java-Script. Полное руководство. /М: Вильямс, 2011
4. James Edwards, Cameron Adams. The JavaScript Anthology. 101 Essential Tips, Tricks & Hacks. / Published by SitePoint Pty. Ltd, 2006.
5. Д. Котеров, А.Костарев. PHP5. / СПб: БХВ-Петербург, 2005

6. Ульман Л. MySQL Руководство по изучению языка. / ДМК Пресс; Питер, 2004

Дополнительная литература

7. С. Айзекс. Dynamic HTML. Секреты создания интерактивных Web-страниц / СПб: БХВ-Петербург, 2001
8. Вивек Шарма, Раджив Шарма. Разработка Web-серверов для электронной коммерции. Комплексный подход. /М: Вильямс, 2001
9. Джерри Брандебау. JavaScript: сборник рецептов/ СПб: Питер, 2001
10. А. Павлов. CGI-программирование. / СПб: Питер, 2000
11. М.Кузнецов, И. Симдянов, С.Гольшев. PHP5. Практика создания web-сайтов. / СПб: БХВ-Петербург, 2005
12. Система управления контентом с открытым исходным кодом. Руководство пользователя для Joomla 1.0.11 Open Source. / <http://www.joomla.ua>, <http://www.joomlaportal.ru>

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

1. Интернет-университет информационных технологий. Web-технологии [электр. ресурс]
<http://www.intuit.ru/studies/courses/485/341/info>
2. Интернет-университет информационных технологий. Основы программирования на JavaScript [электр. ресурс]
<http://www.intuit.ru/studies/courses/1093/132/info>
3. www.citforum.ru/internet - раздел «Internet-технологии» на сайте citforum.ru

10. Методические указания для обучающихся по освоению дисциплины.

Дисциплина предусматривает получение и овладение комплексом необходимых знаний и умений в области создания Web-приложений, современных серверных Web-технологий и профессионального программирования для сети Интернет.

Для более полного понимания целей, задач и практических результатов Web-технологий следует:

- 1) Ознакомиться с дополнительной литературой, особенно с трудами основоположников.
- 2) Ознакомиться, хотя бы поверхностно, с другими подходами создания Web-приложений.
- 3) Попытаться в рамках занятий самостоятельно и полностью выполнить все задания.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

- Denwer набор дистрибутивов (Apache, PHP, MySQL, Perl и т.д.);
- PHP5 с поддержкой GD, MySQL, sqLite;
- Браузер (Mozilla Firefox, Google Chrome, Opera, Safari, Internet Explorer).

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

- Компьютерный класс;
- 11 компьютеров;
- Типы: Pentium IV или выше;
- Проектор

