

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
по дисциплине «Web-технологии и приложения»
для студентов направления подготовки
11.03.02 Инфокоммуникационные технологии и системы связи

Ставрополь 2026

Содержание

Лабораторная работа 1. Язык разметки гипертекста HTML. Основные понятия

Лабораторная работа 2. Форматирование текста. Управление цветом

Лабораторная работа 3. Гиперссылки.

Лабораторная работа 4. Работа со списком.

Лабораторная работа 5. Вставка графических изображений

Лабораторная работа 6. Таблицы HTML-документов

Лабораторная работа 7. Формы в HTML.

Лабораторная работа 8. Технология CSS

Лабораторная работа 1

Тема: Язык разметки гипертекста HTML. Основные понятия

Цель работы: ознакомить студентов с основными понятиями языка HTML, структурой HTML-документа, обязательными метками, комментариями, способами форматирования текста, физическими и логическими стилями, приобретение навыков создания простейших статических Web-документов.

Теоретическая часть

Основы языка разметки гипертекста - HTML

Базовым элементом языка разметки гипертекста является - ТЕГ (дескриптор, маркер). Тег всегда заключен между скобками < > и имеет следующий вид: <ТЕГ параметр1="ЗНАЧЕНИЕ" ... параметрN="ЗНАЧЕНИЕ">

Теги бывают одиночными и контейнерными. *Контейнером* называется пара: открывающий <ТЕГ> и закрывающий </ТЕГ>.

<ТЕГ> Содержимое контейнера </ТЕГ>

Открывающий тег служит для указания программе-браузеру начала какого-либо объекта или задания свойств объектов помещенных в контейнер. Закрывающий тег служит для указания программе-браузеру о конце объекта или окончания применения свойств, заданных в открывающем теге. Параметры (атрибуты) тега задают значения свойств данного объекта или объектов помещенных в контейнер. Значения свойств, содержащие пробелы, берутся в кавычки, в остальных случаях кавычки можно опустить.

HTML документ представляет собой обычный текстовый файл, содержащий маркированный тегами форматирования текст, а так же заданные специальными тегами ссылки на графические и прочие файлы мультимедиа, ссылки на другие документы HTML и ресурсы Internet.

Браузер (Browser) - программа для просмотра Web-страниц.

Структура HTML-документа

Документ HTML начинается открывающим тегом <HTML> и заканчивается за-крывающим тегом </HTML>. Между данной парой контейнерных тегов

располагаются две другие основные части HTML документа: заголовок заключенный в контейнер `<HEAD>...</HEAD>` и тело документа в контейнере `<BODY>...</BODY>`. Таким образом, структура простого HTML документа выглядит примерно так:

Структура HTML-документа

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2//EN" >
<HTML>
<HEAD> Заголовок документа </HEAD>
<BODY> Тело документа </BODY>
</HTML>
```

Объявление <!DOCTYPE>

Элемент `<!DOCTYPE>` должен первым указываться в документе HTML (теоретически). Он сообщает серверу WEB способ обработки документа и то, какие дескрипторы могут находиться на странице, хотя чаще всего он игнорируется браузерами. Поэтому его применение строго не обязательно.

Синтаксис: `<!DOCTYPE HTML "текст""URL">`

Здесь текст определяет версию HTML, а URL позволяет браузерам пользователей загрузить DTD например:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://WWW.W3.ORG/TR/REC-HTML40/strict.dtd">
```

Тэг <HTML>

Тэг `<HTML>` определяет границы документа HTML, ему соответствует конечный тэг `</HTML>`. Между этими двумя тэгами располагается собственно весь документ. Как и `<!DOCTYPE>` тэги `<HTML>` и `</HTML>` - не являются строго обязательными. Но, все-таки, их использование является правилами хорошего тона т.к. браузеры у пользователей могут быть всякие и не известно - насколько корректно они визуализируют такой код.

В дополнение к обязательной структуре настоятельно рекомендуется вставлять различные структурные детали.

Каждый HTML документ должен содержать основную информацию о его происхождении.

Если Вы стремитесь к тому, чтобы люди отыскиали Ваш документ по соответствующим связям, важность предоставления информации о его происхождении становится очевидной. Когда пользователь найдет Ваш документ с помощью, например, поискового ресурса AltaVista, он, вероятнее всего, захочет узнать, к какому виду относится документ. Поэтому каждый файл HTML должен предоставлять самую основную информацию (или связи к информации) о его происхождении и природе. Например, в собрании книгоподобных документов, разделенных на малые файлы, каждый файл должен содержать, по крайней мере, связь к "первой странице" "книги" (home page).

О происхождении документа должна быть представлена, по крайней мере, следующая информация:

Автор документа, имеющий уникальное имя. При этом должна быть задана связь с домашней страницей автора. Если у документа несколько авторов, определите их всех, а также роль каждого из них; например, ведущего автора, редактора, действующего спонсора, а также лиц, формально отвечающих за документ. Дата создания документа или его последней модификации, или и та и другая. Представляемая дата должна быть понятна во всем мире; в частности, название месяца лучше писать словом, а не цифрой. Контекст документа и его статус, например: часть официальной документации компании об одном из ее продуктов, или часть личной информации о хобби автора, или что-то другое. Адрес (URL) документа. Такая информация зачастую чрезмерна, однако она может быть очень полезной, когда кому-то нужна копия именно того документа, который он нашел. Лучше не полагаться на браузер (и пользователя), добавляющих такую информацию, когда сделана копия документа.

Элемент <BODY>

Элемент <BODY> предназначен для выделения той части документа, которая будет визуализирована для пользователя. Он имеет как начальный, так и конечный теги. Начальный тег <BODY> может иметь несколько атрибутов.

Заголовочные тэги

Элемент `<HEAD> </HEAD>` определяет заголовок документа.

BASE - базовый, основной URL. Цель: задание базового URL для относительных URL в документе (например, в атрибутах `HREF` элемента `A`). Этот элемент часто используется для отображения документов. Например: `<BASE href="http://foo.com/index.html">`.

Элемент `BASE` непосредственно не отображается в документе. Основной синтаксис `<BASE HREF="URL">`. Так как в документе допускается только один элемент `BASE`, Вы не можете иметь различные базовые URL в различных частях файла HTML. При отсутствии элемента `BASE` в документе URL самого документа становится базовым в пределах документа.

META - метаинформация

Используется для задания метаинформации (информации о документе), т.е. пар имя/значение, описывающих свойства документа, например, авторство, истечение даты, список ключевых слов и т.д. Типичное отображение. Элементы `META` не влияют на отображение самого документа. Они могут давать некоторый эффект при представлении информации о документе, например, в верхнем окне браузера или в ответе на запрос от поискового средства.

Синтаксис:

`<META NAME=имя элемента метаинформации CONTENT=содержимое информации>` или `<META HTTP-EQUIV=имя элемента метаинформации CONTENT=содержимое информации>`.

Примеры:

`<META NAME=DESCRIPTION CONTENT="An extensive guide to writing HTML 3.2 documents, with ../examples and practical advice.">`

`<META NAME=KEYWORDS CONTENT="structural HTML, logical markup">`.

Тег `META` влияет на индексирование документа, когда он включается в базу данных поискового сервера.

TITLE - "внешний" заголовок (титул)

Используется для задания обязательного "внешнего" заголовка документа.

Титул может выводиться в окне заголовка программы просмотра; в списке результатов поиска, возвращаемых поисковым сервером; в горячем списке, определяемом пользователем; списке истории и т.д.

Основной синтаксис <TITLE>последовательность символов</TITLE>

В TITLE можно использовать escape последовательности, например, < (для <) и ¨ (для д), но никакие теги HTML не разрешены, поэтому Вы не можете задавать в заголовке размеры шрифтов или выделения.

Примеры <TITLE>A study of population dynamics</TITLE> Примечания

Написать хорошее заглавие - очень важно, так как списки результатов поиска, возвращаемые поисковым сервером, могут использовать его.

Комментарии

Файл HTML может содержать комментарии, дающие пояснения для человека, читающего HTML код. Комментарии не влияют каким-либо образом на представление документа, т.е. они игнорируются бро-узером. Вы можете начать комментарии с четырехсимвольной последовательности <!-- (знак "меньше чем", восклицательный знак, два дефиса) и завершить его трехсимвольной последовательностью --> (два дефиса, знак "больше чем"). Например: <!-- Написано Иваном Ивановым -->.

Пример 1

```
<HTML>
<HEAD>
<TITLE> моя первая страница </title>
<META NAME="AUTHOR" CONTENT=" SIDOROV SIDOR ">
<META NAME="KEYWORDS" CONTENT=" Животные, природа, фауна ">
</head>
<BODY>
Здравствуй, это моя первая страница.
<!--<PRE> КОММЕНТАРИЙ </PRE>-->
<BR>
Добро пожаловать! :)
</body>
</html>
```

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задания

Указания к выполнению:

1. Создайте на диске отдельную директорию (папку) для будущей страницы.
2. Откройте текстовый редактор Блокнот (notepad) и наберите в нем структуру HTML-документа, которая приведена в примере 1.
3. Сохраните этот документ, присвоив ему имя *.html
4. Откройте с помощью браузера ваш документ (не закрывайте блокнот, он нам еще пригодится).

Если вы что-то изменили в вашем *.html документе (в Блокноте), то, не забудьте сохранить изменения, а чтобы посмотреть как это выглядит, надо нажать в браузере кнопку ОБНОВИТЬ.

Если изменений не видно, то это значит, что вы где-то что-то неправильно написали, или забыли сохранить документ.

5. Вернитесь к сохраненному в *Блокноте* файлу.
6. Внесите в него следующие изменения: пусть это будет ваша первая страничка. Укажите в ней ваши фамилию, имя, ФИО родителей, братьев и сестер, свои увлечения. Используйте для этого форматирование абзацев.
7. В строке <title> укажите: *"Домашняя страничка (ваше имя и фамилия)"*.
8. Сохраните файл как *page2.htm*.
9. Просмотрите результат в браузере, при необходимости отредактируйте файл при помощи Блокнота.

Содержание отчета

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- вариант вашего задания;

- распечатку страниц выполненного задания;
- выводы по проделанной работе.

Защита работы

Защита работы заключается:

- в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- демонстрации преподавателю выполненного задания;
- в ответах на контрольные вопросы по лабораторной работе;
- в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

1. Охарактеризуйте язык тегов HTML.
2. Какова структура HTML-документов.
3. Перечислите обязательные метки и охарактеризуйте их.
4. Дайте определения следующих понятий: гипертекст, гиперссылка, Web-сайт, URL-адрес в Интернете, фрейм апплет скрипт, браузер элемент, тэг, атрибут.

Лабораторная работа 2

Тема: Форматирование текста. Управление цветом

Цель работы: научиться форматировать шрифт и текст на Web-странице, научиться создавать цветной фон Web-страницы, использовать шрифт различного цвета.

Теоретическая часть

Форматирование текста

Текст — единственный объект Web-страницы, который не требует специального определения. Иными словами, произвольные символы интерпретируются по умолчанию как текстовые данные. Для форматирования текста существует большое количество элементов. Форматировать текст можно с помощью традиционных элементов: выделять фрагменты курсивом, полужирным

шрифтом, выбирать шрифт, размер и цвет шрифта, выравнивать текстовые фрагменты. Если автору не хватает простых вариантов форматирования текста, он может прибегнуть к таблицам стилей, которые существенно расширят возможности языка HTML по форматированию. Рассмотрим стандартные элементы языка HTML, позволяющие форматировать текст.

Форматирование абзаца

Элемент <P> Этот элемент задает один из способов разбиения текста на абзацы. Он может иметь вложенный атрибут align, который указывает отступ left, center, right, justify. Каждый следующий абзац игнорирует, заданное для предыдущего абзаца значение align.

Синтаксис:

<P align=отступ> Текст абзаца </P>

Элемент абзаца разделяет фрагменты текста вертикальным отступом. Он позволяет использовать только начальный тэг, так как следующий элемент </p> обозначает не только начало следующего абзаца, но и конец предыдущего. В тех случаях, когда по смыслу необходимо обозначить завершение абзаца, можно использовать и конечный тэг. Вместе с элементом абзаца можно использовать и атрибут выравнивания **align**:

- <p align="left"> – выравнивание по левому краю;
- <p align="right"> – выравнивание по правому краю;
- <p align="center"> – выравнивание по центру;
- <p align="justify"> – выравнивание по ширине. Выравнивание по ширине – опасный параграф. Он не работает в старых версиях браузеров (т.е. текст выравнивается в таком случае по левому краю документа).

Пример 1

```
<HTML>
<HEAD>
<TITLE> Использование различных типов выравнивания </title>
</head>
<BODY>
<P align="justify"> Элемент абзаца разделяет фрагменты текста вертикальным отступом.
Вместе с элементом абзаца можно использовать и атрибут выравнивания align: </p>
<P align="left"> aligns="left"–выравнивание по левому краю; </p>
<P align="center"> align="center" – выравнивание по центру; </p>
```

```
<P align="right"> align="right"—выравнивание по правому краю; </p>
<P align="justify"> align="justify" – выравнивание по ширине.</p>
</body>
</html>
```

**
**

Элемент, обеспечивающий принудительный переход на новую строку. Он имеет только начальный тэг. В месте его размещения строка заканчивается, а оставшийся текст печатается с новой строки.

Пример 2

```
<HTML>
<HEAD>
<TITLE> Принудительный перенос строки </title>
</head>
<BODY>
Уронили мишку на пол,
<BR> Оторвали мишке лапу,
<BR> Все равно его не брошу
<BR> Потому, что он хороший.
</body>
</html>
```

Если таким образом расставить элемент **
** в этом стихотворении, то в экране браузера мы увидим стандартным образом написанное четверостишие на четырех строках.

<NOBR>...</nobr>

Этот элемент по своему действию является прямой противоположностью предыдущему. Текст, заключенный между его тегам, будет выведен в одну строку. Длинная строка не уместится на экране, и для ее просмотра придется использовать горизонтальную полосу прокрутки.

<pre>...</pre>

Элемент для обозначения текста, отформатированного заранее (*preformatted*). Подразумевается, что текст будет выведен в том виде, в каком он был подготовлен пользователем. Текст, заключенный между метками **<PRE>** и **</pre>** (от английского *preformatted* — предварительно форматированный), выводится браузером на экран как есть — со всеми пробелами, символами табуляции и конца строки. Во всех других случаях браузер игнорирует эти символы. Возможен и обратный эффект: если пользователь введет текст как одну длинную

строку, то она не будет разорвана браузером, а уйдет за границы окна программы. В этом смысле элемент *PRE* работает так же, как элемент *NOBR*. По умолчанию для отформатированного заранее текста выбирается моноширинный шрифт. Этот элемент удобно использовать для демонстрации листингов программ или для вывода текстовых документов, переформатирование которых может привести к искажению их смысла.

<BLOCKQUOTE> ... </blockquote>

Текст, заключенный между метками *<BLOCKQUOTE>* и *</blockquote>*, выводится браузером на экран с увеличенным левым полем (с отступом).

<CENTER>...</center>

Элемент для центрирования текста, а точнее, любого содержимого. Не является общеупотребительным. В тех случаях, когда это возможно, вместо него используют атрибут *align = "center"*.

<DIV>...</div>

Элемент, похожий на предыдущий. Он позволяет выравнивать содержимое по левому краю, по центру или по правому краю. Для этого стартовый тег должен содержать соответствующий атрибут:

- *align = "left"*
- *align = "center"*
- *align = "right"*

Форматирование шрифта HTML допускает два подхода к шрифтовому выделению фрагментов текста – логическое и физическое форматирование.

В группу тегов логического форматирования входят теги, отображающие на экране монитора элементы документа, таким образом, как установлено по умолчанию в спецификации языка разметки HTML. Переопределить их параметры или свойства нельзя, за исключением ситуаций использования стилевых шаблонов CSS и обособления тегами физического форматирования. Результат действия разных тегов логического форматирования визуально может совпадать, ибо основное их предназначение заключается в логическом выделении отдельных элементов HTML.

Теги физического форматирования позволяют разработчику визуально изменять вид текста, делая его жирным или наклонным, варьируя его параметры и значения. Под физическом стилем принято понимать прямое указание браузеру на модификацию текущего шрифта.

...

Выделение текста полужирным шрифтом (от слова *Bold*), - очень популярный элемент. Использование полужирного шрифта – прием, позаимствованный из текстовых редакторов. Т.е. текст находящийся внутри тега будет жирным.

<I>...</i>

Выделение текста курсивом (от слова *italic*). Т.е. текст, находящийся внутри тега *под наклоном*

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задания

Указания к выполнению задания 1

1. Создайте новую Web-страницу в текстовом редакторе Блокнот.
2. В элементе `<title>` укажите название странички "Задание 1. Форматирование текста на Web-странице по образцу".
3. Отформатируйте следующий текст согласно указаниям таблицы.

Текст	Применить следующие форматирование
Автомобили	Заголовок H1
Легковые	Заголовок H2
Отечественные	выравнивание слева, полужирный шрифт, цвет шрифта зеленый
Импортные	выравнивания справа, полужирный шрифт, цвет шрифта оранжевый
Тестдрайв	выравнивание по центру, полужирный курсив
Салоны продаж	Выравнивание по центру, размер шрифта 3, цвет шрифта красный
Стихотворение	цвет шрифта синий, выравнивание по центру
Уронили мишку на пол,	Выравнивание по левому краю, размер шрифта 4

Оторвали мишке лапу. Все равно его не брошу - Потому что он хороший.	
--	--

Указания к выполнению задания 2

1. Создайте новую Web-страницу в текстовом редакторе Блокнот.

2. В элементе `<title>` укажите название странички "Задание 2. Форматирование текста на Web-странице по образцу".

3. Напишите HTML-файл, чтобы при его выполнении получился текст следующего формата:

Элементы форматирования текста

Здесь находится цитата...

• Фрагменты текста можно выделять: **жирным** или *наклонным* шрифтом. Кроме того, можно включать в текст фрагменты с фиксированной шириной символа (имитация пишущей машинки).

• Текст может быть или подчеркнутым.

• Можно также включать элементы, создающие эффект нижнего индекса или верхнего индекса.

• Использование увеличения шрифта – BIG

• Использование уменьшения шрифта – SMALL

• Кроме физических существуют также логические стили

• Использование акцента – EM

• Использование сильного акцента – STRONG

• Фрагмент исходного текста CODE

• Использование образца – SAMP

• Текст, вводимый с клавиатуры – KBD

• Значение переменной – VAR

• Использование элемента CITE

• Так выглядит формат адреса (элемент ADDRESS)

• Использование элемента CENTER

• Использование элемента DIV

- Элемент BR обеспечивает разрыв строки

Содержание отчета и его форма

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- вариант вашего задания;
- распечатку страниц выполненного задания;
- выводы по проделанной работе.

Защита работы

Защита работы заключается:

- в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- демонстрации преподавателю выполненного задания;
- в ответах на контрольные вопросы по лабораторной работе;
- в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

1. Дайте характеристику форматированию абзаца.
2. Дайте характеристику форматированию шрифта.
3. Логические стили в HTML - документах.
4. Комментарии в HTML-документах.
5. Escape последовательности.
6. Что такое цветовая гамма HTML-документа
7. Сочетание цветов в документе

Лабораторная работа 3

Тема: Гиперссылки.

Цель работы: познакомиться с основными принципами построения гипертекстовых ссылок.

Теоретическая часть

Гипертекстовые ссылки

HTML предлагает множество условных оборотов для текстовых и структурированных документов, но что отличает его от большинства других языков разметки - его возможности разметки гипертекста и интерактивных документов. Ссылки связывают один ресурс Web с другим. Несмотря на простоту, ссылки стали основным залогом успеха Web.

Гипертекст (Hyper Text) - это многомерный текст, т.е. такая организация документов, при которой один документ или текст может включать в себя разнонаправленные ссылки. Эти ссылки, называемые гипертекстовыми ссылками или гиперссылками (Hyper Text links, hyperlinks), позволяют просматривать документ в любом необходимом порядке. Объединение одномерных текстов с включенными в них гиперссылками называется гипертекстом.

Самым простым примером гипертекста является система контекстной помощи Microsoft Windows.

Любая гиперссылка состоит из двух частей: *указателя ссылки* ("якоря" - от англ. anchors) и *адреса ресурса*, которым может быть любым ресурсом Web (например, изображением, видеоклипом, звуковым файлом, программой, документом HTML, элементом в документе HTML и т.д.), на который необходимо осуществить переход.

Внешне отличить гиперссылку от обычного текста очень просто: при наведении курсора мыши на ссылку указатель принимает вид руки с указательным пальцем, как бы показывающим, что этот текст содержит гиперссылку. Сама ссылка подчеркивается (в случае если указателем является текст).

В качестве указателя может выступать текст (отдельное слово, фразы и даже целые страницы текста) и графические изображения. В ряде случаев возможно объединение графики и текста в рамках единого указателя ссылки.

Гипертекстовые ссылки являются ключевым компонентом, делающим WEB привлекательным для пользователей. Добавляя гипертекстовые ссылки (далее - ссылки), вы делаете набор документов связанным и структурированным, что позволяет пользователю получать необходимую ему информацию максимально быстро и удобно. Ссылки имеют стандартный формат, что позволяет браузеру интерпретировать их и выполнять необходимые функции (вызывать методы) в зависимости от типа ссылки.

Ссылки могут указывать на другой документ, специальное место данного документа или выполнять другие функции, например запрашивать файл по FTP-протоколу для отображения его браузером.

URL может указывать на специальное место по абсолютному пути доступа, или указывать на документ в текущем пути доступа, что часто используется при организации больших структурированных WEB-сайтов.

URL

HTML использует URL (Uniform Resource Locator) для представления гипертекстовых ссылок и ссылок на сетевые сервисы внутри HTML-документа. Первая часть URL (до двоеточия) описывает метод доступа или сетевой сервис. Другая часть URL (после двоеточия) интерпретируется в зависимости от метода доступа. Обычно, два прямых слэша после двоеточия обозначают имя машины:

Uniform Resource Locator имеет следующий формат:

method://servername:port/pathname#anchor.

Структура гиперссылки

Указатель ссылки описывается тегом **<A>**, а адрес перехода реализован с помощью параметра **HREF**, значением которого является путь к тому или иному интернет-ресурсу.

<A>...

Если файл находится в том же каталоге, что и документ, на который сделана ссылка, то полный путь к документу указывать необязательно. Достаточно использовать сокращенные версии адресов, называемые относительными адресами.

Чаще всего используются следующие шаблоны:

- *текст* `<A href="Адрес ссылки"> текст для щелчка </a>`
- `<A href="Адрес ссылки"> <IMG src="Ссылка на рисунок"> </a>`

Первый шаблон применяется в том случае, когда гиперссылка встречается в тексте. Атрибут *href* может указывать на ресурс **Internet**, файл на локальном диске или метку внутри текущей страницы. Текст, расположенный внутри элемента `<A>`, представляет собой видимую часть гиперссылки. Именно на нем должен щелкнуть пользователь, чтобы осуществить переход. Браузер выделяет этот фрагмент цветом, а после использования гиперссылки меняет цвет, чтобы обеспечить подсказку.

link - определяет цвет выделенного элемента текста, при нажатии на который происходит переход по гипертекстовой ссылке.

– **link** = `"#FF0000"` - цвет гипертекстовой ссылки. Насыщенность красным - *FF* (255), зеленым и синим - *00* (ноль). Результат - красный цвет.

– **vlink** - определяет цвет ссылки на документ, который уже был просмотрен ранее.

– **alink** - определяет цвет ссылки в момент, когда на нее указывает курсор мыши и нажата ее правая кнопка, то есть непосредственно перед переходом по ссылке.

Пример 1

```
<HTML>
<HEAD>
<TITLE> Использование текстовой гиперссылки</TITLE>
</HEAD>
<BODY>
<Любая гиперссылка состоит из двух частей:
<A HREF ="1.html"> указателя ссылки </a>
("якоря" - от англ. anchors) и
<A HREF="2.html"> адреса ресурса</a>,
которым может быть любым ресурсом Web, на который необходимо осуществить
```

переход.
</BODY>
</HTML>

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задания

Задание 1. Создание простейшей гиперссылки

Преобразуйте страницы в простой сайт, связав отдельные части с помощью гиперссылок. Необходимо использовать все виды гиперссылок.

Обязательные страницы:

- Об авторе.
- Мой ВУЗ - о СКФУ.
- Моя группа.
- Моя будущая профессия.
- Мои увлечения или хобби.

• На страничке *Моя будущая профессия* создайте в конце документа гиперссылку для перехода в начало этого же документа (якорь).

• Откройте в браузере страничку *Об авторе* и убедитесь, что теперь можно перейти с любой странички на любую другую из созданных.

Задание 2. Цвет гиперссылок

• На страничках, созданных в задании 1, в тэге *<BODY>* определите цвет гиперссылок:

- все гиперссылки на странице - белые;
- активные гиперссылки - красные;
- посещенные гиперссылки - серые.

• Откройте в браузере первую страничку и, переходя по ссылкам со странички на страничку, убедитесь, что цвет гиперссылок задан верно.

Содержание отчета и его форма

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- ☐ вариант вашего задания;
- ☐ распечатку страниц выполненного задания;
- ☐ выводы по проделанной работе.

Защита работы

Защита работы заключается:

- ☐ в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- ☐ демонстрации преподавателю выполненного задания;
- ☐ в ответах на контрольные вопросы по лабораторной работе;
- ☐ в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

1. Понятие гипертекста
2. Понятие гиперссылки.
3. Структура гиперссылки
4. Относительный адрес.
5. Абсолютный адрес.
6. Внешние и внутренние гиперссылки.

Лабораторная работа 4

Тема: Работа со списком.

Цель работы: научиться создавать списки на Web-странице, научиться размещать графические изображения на Web-странице.

Работа со списками

Очень важный элемент Web-страничек – *списки*. Использование списков в качестве средства представления информации объясняется следующим:

1. Информация в виде списка позволяет создавать большие массивы данных на отдельные четкие фрагменты, которые человеку гораздо удобнее воспринимать, чем весь поток целиком.
2. Списки позволяют создавать понятную и логичную внутреннюю структуру информационных данных ориентироваться в которой просто и удобно.
3. Использование списков удобно для отображения определенных пошаговых и прочих последовательных процессов. Список отличается от обычного текста тем, что пользователю не надо думать о нумерации его пунктов: эту задачу программа берет на себя. Если список дополняется новыми пунктами или укорачивается, нумерация корректируется автоматически. В случае нумерованных списков программа ставит перед каждым пунктом маркеры: кружки, прямоугольники, ромбы или другие изображения. В результате список принимает удобочитаемый вид.

Спецификация HTML предусматривает три основных типа списков:

- маркероанные списки;
- нумерованные списки;
- списки определений.

Маркированный (нумерованный - Unordered List) список.

Маркероанный список представляет собой нумероанный или неупорядоченный перечень элементов, для заголовка которых используются специальные маркеры. В качестве маркеров выступают специальные символы, называемые буллетами (bullets). Текст, расположенный между тегами `<UL>` и `</ul>`, воспринимается как нумероанный список.

Элемент `<UL>`, используется с целью задания нумероанных списков, имеет атрибут `type=ciRcle,squaRe`, или `disc` для задания вида маркера. Элемент

 включает в себя дополнительный элемент , который задает элементы списка.

Синтаксис:

<UL type=circle >

 элемент списка

 элемент списка

Для элемента *UL* возможными значениями атрибута *type* являются *disc*, *square* и *circle*, где

- **disc** – маркер в виде закрашенного круга
- **circle** – маркер в виде незакрашенного круга
- **square** – маркер в виде закрашенного квадрата

Пример 2

```
<UL>
<LI> <font color = red>красный;</font>
<LI> <font color = 900090000000>желтый;</font>
<LI> <font color = green>зеленый;</font>
</ul>
<UL type=circle>
<LI> <font color = red>красный;</font>
<LI> <font color = 900090000000>желтый;</font>
<LI> <font color = green>зеленый;</font>
</ul>
<UL type=square>
<LI> <font color = red>красный;</font>
<LI> <font color = 900090000000>желтый;</font>
<LI> <font color = green>зеленый;</font>
</ul>
```

После каждого пункта списка осуществляется перенос строки, причем без указания каких-либо структурных тегов. Если заголовок списка размещается между тегом *UL* и *LI*, то он будет отделенот начала самого списка одним переносом строки и отступом от левого края документа (как и все пункты всписка). При указании заголовка перед тегом *UL* браузер создает двойной перенос и размещает текст заголовка левее пунктов списка. Отличительной особенностью маркерованных списков является возможность использования вместо стандартных HTML-маркеров свои собственные графические

изображения. Для указания браузеру того, что в качестве маркера будет использовано графическое изображение, внутри тега *UL* вместо указателя элемента *LI* размещается обыкновенная HTML-конструкция описания графических объектов.

Пример 3

```
<UL>
<IMG src = marker.gif> <font color = red>красный</font><br><br>
<IMG src = marker.gif> <font color = 9000900000000>желтый</font><br><br>
<IMG src = marker.gif> <font color = green>зеленый</font><br><br>
</UL>
```

Нумерованные списки

Нумерованный список часто называется упорядоченным и представляет собой определенную последовательность элементов. Нумерованные списки (Ordered List) отмечают тегами **... ** внутри которых располагаются пункты перечня информационных данных, за указание которых отвечает специальный тег *LI*. Чтобы программа правильно расставила номера вместо символов, выделяющих новый элемент, используются арабские или римские числа или буквы латинского алфавита.

Синтаксис:

<OL type=1 start=1 >

**** элемент списка

**** элемент списка ****

Тег составления нумерованных списков *OL* может включать атрибуты *type*, *start* и *compact*. Способ нумерации задается при помощи атрибута **type**.

Тип	Пояснение	Вид
type="1"	арабские цифры	1, 2, 3, ...
type="i"	римские цифры в нижнем регистре	i, ii, iii, ...
type="I"	римские цифры в верхнем регистре	I, II, III, ...
type="a"	латинские буквы нижнего регистра	a, b, c, ...
type="A"	латинские буквы верхнего регистра	A, B, C, ...

Типом нумерации, используемым по умолчанию, является *type="I"*, поэтому при создании нумерованных списков такого типа необязательно создавать значения атрибута *type* (сам перпметр также можно опустить).

Пример 4

```
<OL>
<li> красный;</li>
<li> желтый;</li>
<li> зеленый.</li>
</ol>
<OL type="I">
<li> красный;</li>
<li> желтый;</li>
<li> зеленый.</li>
</ol>
<OL type="A">
<li> красный;</li>
<li> желтый;</li>
<li> зеленый.</li>
</ol>
```

Возможно смешанное использование различных типов нумерации в пределах одного списка. Однако браузеры по-разному реагируют на такое смешение стилей.

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задания

Задание 1. Создание нумерованных и нenumерованных списков.

1. Создайте новую Web-страницу в текстовом редакторе Блокнот.
2. В элементе title укажите название странички "Задание 3. Создание нумерованных и нenumерованных списков на Web-странице по образцу".
3. Отформатируйте следующий текст согласно указаниям (даны в скобках курсивом).

Солнце должно быть: (Шрифт размером 4, красного цвета, выравнивание по центру)

1. *Теплым.*
2. *Круглым.*
3. *Желтым.*

Снег должен быть: (Шрифт размером 10, синего цвета, выравнивание по левому краю)

А. Белым.

В. Холодным.

С. Пушистым.

(Шрифт обоих, списков черный, размер 3).

Задание 2. Создайте маркированный список, содержащий фамилии ваших любимых актеров, создайте нумерованный список, перечисляющий цвета радуги, создайте список определений, содержащий название и описание породы собак.

Задание 3. Рецепт приготовления эликсира создайте в виде вложенного списка, применяя вложение маркированного списка в нумерованный.

1. Создать 3-уровневый список.

- уровень (верхний) - маркированный список стран;
- уровень (средний) - нумерованный список автомобильных фирм;
- уровень (нижний) - маркированный список автомобилей (количество элементов на каждом из уровней - не меньше 2).

Создать список определений (любой, количество элементов не меньше 5).

Создать нумеруемый список в виде меню (MENU).

2. Создать 3-уровневый список.

- уровень (верхний) - нумерованный список стран;
- уровень - маркированный список фирм, производящих принтеры;
- уровень - нумерованный список принтеров (количество элементов на каждом из уровней - не меньше 2).

Создать список определений (любой, количество элементов не меньше 5).

Создать нумерованный список в директории (DIR).

3. Создать 3-уровневый список.

- уровень (верхний) - маркированный список стран;

- уровень - нумерованный список фирм, производящих мониторы;
- уровень - нумерованный список мониторов (количество элементов на каждом из уровней - не меньше 2).

Создать список определений (любой, количество элементов не меньше 5).

Создать нумеруемый список в виде меню (MENU).

Содержание отчета и его форма

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- ☐ вариант вашего задания;
- ☐ распечатку страниц выполненного задания;
- ☐ выводы по проделанной работе.

Защита работы

Защита работы заключается:

- ☐ в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- ☐ демонстрации преподавателю выполненного задания;
- ☐ в ответах на контрольные вопросы по лабораторной работе;
- ☐ в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

1. Перечислите основные типы списков в HTML-документах.
2. Охарактеризуйте маркированный (нумерованный) список.
3. Дайте характеристику нумерованных списков.
4. Дайте характеристику вложенным спискам.
5. Списки определений в HTML-документах.

Лабораторная работа 5

Тема: Вставка графических изображений

Цель работы: научиться размещать графические изображения на Web-странице.

Вставка графических изображений

Чтобы вставить изображение, нужно только иметь это самое изображение в формате *GIF* (файл с расширением *.gif) или *JPEG* (файл с расширением *.jpg или *.jpeg) и одну строчку в HTML-тексте.

**** - элемент для создания ссылки на графический файл (image). Он не содержит конечного тэга - вся необходимая информация задается при помощи атрибутов. Этот элемент является универсальным: с его помощью можно использовать изображения в гиперссылках, вставлять картинки в таблицы, просто размещать рисунки на Web-странице, решать задачи дизайна и т. д.

Необходимым атрибутом является **SRC** - указатель на графический файл:

SRC = "Ссылка на файл".

Допустим, нам нужно включить в документ изображение, записанное в файл picture.gif, находящееся в одном каталоге с HTML-документом. Тогда строчка будет вот такая:

Очень полезным атрибутом является **ALT**. Он позволяет выводить текст в тех местах, где должны располагаться рисунки. Страница может загружаться достаточно долго, и пока графические файлы на "подходе", пользователь должен видеть, какие изображения он сможет получить.

Пример 5

```
<HTML>
<HEAD>
<TITLE>Вставка изображения</title>
</head>
<BODY>
<IMG SRC="picture.gif" ALT="Картинка">
```

```
</body>  
</html>
```

Встретив такую метку, браузер покажет на экране текст "Картинка" и начнет загружать на его место картинку из файла *picture.gif*. Атрибут *ALT* может оказаться необходимым для старых браузеров, которые не поддерживают изображений, а также на случай, если у браузера отключена автоматическая загрузка изображений (при медленном подключении к Интернет это делается для экономии времени).

Файл, содержащий изображение, может находиться в другом каталоге или даже на другом сервере (обратите внимание на рисунок в строку адреса). В этом случае необходимо указать его полное имя (путь в файлу):

```

```

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задания

Задание 1. Размещение графики на Web-странице

1. Создайте новую Web-страницу в редакторе **Блокнот**.
2. В элементе *<TITLE>* укажите название странички «Задание 6. Размещение графики на Web-странице».
3. Если в тэге *BODY* не указывать цвет странички, то по умолчанию фон будет белым. Сделайте цвет фона странички оранжевым.
4. Разместите на страничке любой рисунок. Для размещения изображения вам требуется указать путь к файлу в элементе *IMG*. Чтобы упростить описание пути к этому графическому файлу, скопируйте этот файл в ту же папку, в которой будет сохранена Web-страничка. Если файл находится не в одной папке с Web-страничкой, то необходимо указать относительный путь к этому файлу.
5. Сделайте подпись к рисунку с помощью атрибута *alt*.

6. Создайте рамку вокруг рисунка шириной 2 пиксела.
7. Над рисунком поместите заголовок самого большого размера и выровняйте его по центру страницы.

Задание 2. Изменение размеров изображения на Web-странице.

1. Создайте новую Web-страницу в редакторе **Блокнот**.
2. В элементе *TITLE* укажите название странички «Задание 7. Изменение размеров изображения на Web-странице».
3. Пусть фон странички будет зеленого цвета.
4. Определите размер выбранного вами изображения.
5. Вставьте это изображение на страничку, задав его размеры: ширина 100, высота 100 пикселей.
6. Просмотрите в браузере полученную страничку. Обратите внимание на следующие вещи:
 - В связи с тем, что вы непропорционально изменили размеры изображения, произошло искажение рисунка;
 - Так как вы увеличили размер изображения, произошла потеря качества рисунка (обратите внимание на края изображения);
 - Для размещения изображений на Web-страничке, лучше использовать прозрачный фон рисунка.

Задание 3. Использование рисунка в качестве гиперссылки

На страничках *page1.htm*, *page2.htm* и *page3.htm*, вместо текстовых гиперссылок используйте рисунки.

Для этого:

- Вставьте на первую страничку одно изображение и создайте гиперссылку для перехода на вторую страницу:

```
< A href = "page2.htm" IMG src = "*.gif" > ... </a>
```

самостоятельно вставьте другое изображение для перехода с первой странички на третью;

- задайте бордюр обоим изображениям, равным 2;

- откройте в браузере первую страничку и, переходя по ссылкам со странички на страничку, убедитесь, что переходы работают верно.

- Самостоятельно вставьте изображения на страницы *page2.htm* и *page3.htm* и задайте соответствующие гиперссылки.

- Откройте в браузере первую страничку и, переходя по ссылкам со странички на страничку, убедитесь, что кнопки на всех страницах работают верно.

Содержание отчета и его форма

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- ☐ вариант вашего задания;
- ☐ распечатку страниц выполненного задания;
- ☐ выводы по проделанной работе.

Защита работы

Защита работы заключается:

- ☐ в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- ☐ демонстрации преподавателю выполненного задания;
- ☐ в ответах на контрольные вопросы по лабораторной работе;
- ☐ в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

1. Роль графики. Характеристика графических стандартов
2. Вставка графических изображений в HTML-документ.
3. Использование карт изображений

Лабораторная работа 6

Тема: Таблицы HTML-документов

Цель работы: познакомиться с основными приемами создания таблиц на языке гипертекстов, рассмотреть основные теги и атрибуты, научиться создавать, форматировать, заполнять таблицы, проектировать структуру таблицы.

Теоретическая часть

Одним из наиболее мощных и гибких средств представления информационных данных в HTML по праву являются таблицы.

Таблицы, изначально избранные в качестве визуального способа представления данных, сейчас имеют гораздо более важную функцию - управление структурой HTML-документа в целом. Удобство размещения данных в таблице неоспоримо, а преимущества перед другими средствами представления информации (например, списки) позволяют считать таблицы основополагающим структурным элементом любого HTML-документа.

Таблицы являются очень удобным средством форматирования данных на Web-странице. Основное удобство заключается в том, что браузер берет на себя заботу о прорисовке рамки таблицы. Размер рамки может быть автоматически согласован с размером окна просмотра в браузере и, разумеется, с размером находящихся в ячейках таблицы строк текста и рисунков. Кроме этого, таблицы позволяют решать чисто дизайнерские задачи: выравнивать части страницы друг относительно друга, размещать рядом рисунки и текст, управлять цветовым оформлением и т. д. Контурные таблицы показывать не обязательно, поэтому тег `<TABLE>` применим для размещения текстовых и графических компонент в разных местах страницы. При создании таблиц используется принцип вложения: внутри основного элемента таблицы (`TABLE`) создается ряд элементов, определяющих строки (`TR`), а внутри этих элементов размещаются элементы для описания каждой ячейки в строке (`TD`, `TH`).

Основные методы создания таблиц

Для описания таблиц используется тег `<TABLE>`.

Тег <TABLE>, как и многие другие, автоматически переводит строку до и после таблицы.

Он позволяет задавать общие свойства таблицы и отделяет ее структуру от остальной части Web-страницы. Если начало и конец таблицы не определены, то никакие табличные команды не будут работать.

Рассмотрим параметры этого элемента.

Параметр align

Таблицу можно выравнивать по горизонтали при помощи атрибута *align*. Данный параметр устанавливает тип выравнивания всей таблицы относительно ширины HTML-документа.

- *align="left"* - влево;
- *align="center"* - по центру;
- *align="right"* - вправо.

Параметры width и height

Параметры *width* и *height* отвечают соответственно за ширину и высоту таблицы. Значение этих параметров может быть указано в процентах или пикселах:

Пример 1

```
<TABLE WIDTH="50%" HEIGHT="250">
```

Где *width = "50%"* задает таблицу шириной 50% страницы; *height = "250"* задает ширину таблицы, равной 250 пикселей.

В случае указания ширины или высоты в пикселах таблица будет иметь фиксированные размеры. Такая таблица при изменении размеров окна браузера останется неизменной. При использовании в качестве единицы измерения процентов ширина/высота таблицы будет варьироваться в зависимости от размеров окна браузера. В этом случае размер ячеек таблицы будет уменьшаться/увеличиваться пропорционально заявленному значению в процентах. Содержимое таблицы будет перемещаться в окне браузера соответственно изменению его размеров.

Для управления видом рамки используются два атрибута. Дело в том, что браузер создает изображение рамки, имитируя ее трехмерность (выпуклость). На рамке можно различить фронтальную и боковую наклонную грани. Кроме этого, имитируется различие в их освещенности.

Параметр border

Параметр *border* задает ширину рамки. Если рамка не задана, то получим таблицу без рамок, тот же результат дает `<TABLE border="0">`. Также как и ширину можно задать в процентах и пикселях.

border = "Ширина в пикселях"

Параметр bordercolor

Параметр *bordercolor* предназначен для определения цвета рамки вокруг таблицы (поддерживается только Internet Explorer). Значение указывается в шестнадцатеричном формате или в виде наименования (например, *green*).

Параметр cellpadding

Параметр *cellpadding* задает расстояние между соседними ячейками таблицы по вертикали и горизонтали. Расстояние между соседними ячейками (как по вертикали, так и по горизонтали) всегда будет одинаково, т. к. HTML не позволяет назначать разные значения в пределах данного параметра.

cellpadding = Ширина в пикселях

Если значение этого атрибута равно нулю, рамка получается тонкой, заостренной. Если данный параметр не задан, по умолчанию задается величина, равная 2 пикселям. Атрибут полезен для размещения текста и графики именно там, где нужно.

Параметр cellspacing

Для всех ячеек таблицы можно задать размер пустого пространства между содержимым таблицы и ее границами. Для создания и регулирования отступа между рамкой ячейки и ее содержимым (по вертикали и горизонтали) используется параметр *cellpadding*.

cellpadding = Число пикселей

Задание этого атрибута делает ячейки больше. Между рамкой таблиц и данными всегда сохраняется определенное расстояние. В некоторых случаях это позволяет улучшить восприятие таблицы, сделать текст в ячейках легко читаемым.

При значении параметра *cellpadding*, равном нулю, содержимое ячейки будет вплотную прижато к рамке таблицы, что ухудшит восприятие информации (это утверждение еще более актуально для таблиц с прозрачной рамкой, т. е. значением *border="0"*, - в этом случае текст соседних ячеек сольется друг с другом).

Группировка данных. Колонтитулы таблицы. Вложенные таблицы

Группировка данных

При построении таблиц мы можем легко задать единый тип выравнивания для отдельной ячейки и даже целого табличного ряда. Однако гораздо, чаще может возникнуть необходимость в едином выравнивании информационных данных конкретного столбца. *Столбец в таблице* - это последовательность ячеек, располагающихся в разных рядах.

Стандартными средствами HTML нам пришлось бы задавать один и тот же тип выравнивания для отдельно взятой ячейки, формирующей нужный столбец:

```
<TABLE>
<TR>
<TD align="right">Ячейка 1 с выравниванием вправо</td>
<TD align="center">Ячейка 1 с выравниванием по центру</td>
</tr>
<TR>
<TD align="right">Ячейка 2 с выравниванием вправо</td>
<TD align="center">Ячейка 2 с выравниванием по центру</td>
</tr>
</table>
```

При работе с браузером Internet Explorer, труд разработчика электронного документа может быть значительно облегчен за счет таких тегов, как *<col>* и *<colgroup>*.

Теги *<col>* и *<colgroup>* предназначены для определения свойств отображения табличных данных, сгруппированных по конкретному признаку.

Они оба могут иметь параметр *span*, задающий количество соседних столбцов, и параметр *align*, становливающий единый тип выравнивания для выбранного столбца (возможные значения: по левому краю, по правому краю, по центру; формат записи аналогичен тегам `<TD>` и `<TH>`).

Помимо этого тег `<colgroup>` может содержать дополнительный параметр вертикального выравнивания данных - *valign* (возможные значения: по верхнему краю, по нижнему краю, по середине; формат записи аналогичен тегам `<TD>` и `<TH>`).

Разница между тегами `<col>` и `<colgroup>` заключается в условном объединении (группировке) взаимосвязанных данных отдельно взятого столбца таблицы.

Пример 12

```
<HTML>
<HEAD>
<TITLE>Группировка данных таблицы</title>
</head>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LEFTMARGIN="40" RIGHTMARGIN="40"
MARGINWIDTH="40">
<TABLE align="center" border="2" cellspacing="0"
cellpadding="5" width="100%" height="200">
<COLGROUP align="center" span="2">
<COLGROUP align="right" span="2">
<TR>
<TD>Ячейка 1</td>
<TD>Ячейка 2</td>
<TD>Ячейка 3</td>
<TD>Ячейка 4</td>
</tr>
<TR>
<TD>Ячейка 5</td >
<TD>Ячейка 6</td>
<TD>Ячейка 7</td>
<TD>Ячейка 8</td>
</tr>
</table>
</body>
</html>
```

Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4
Ячейка 5	Ячейка 6	Ячейка 7	Ячейка 8

Рисунок 6. Результат работы пример 12

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задания

Задание 1. Создание таблицы по образцу

- В элементе `<TITLE>` укажите название странички "Задание 1.

Создание таблицы по образцу"

- Создайте таблицу, как показано на рисунок 8.

Один	Два	Три
Четыре	Пять	Шесть

Рисунок 8. Образец таблицы

- Текст в первой строке выровняйте по центру по горизонтали и по вертикали.
- Текст во второй строке выделите жирным шрифтом.

Задание 2. Создание таблицы по образцу.

- В элементе `<TITLE>` укажите название странички "Задание 2.

Создание таблицы по образцу"

- Создайте HTML-документ, соответствующий следующему образцу, рисунок 9.

Примеры таблиц:

Стандартная таблица

Заголовок таблицы

Заголовок 1	Заголовок 2
Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

Объединение ячеек

Заголовок 1	Заголовок 2	Заголовок 3
Ячейка 1		Ячейка 2
		Ячейка 3
		Ячейка 4
		Ячейка 5

Вложенная таблица

Ячейка 1.1.1	Ячейка 1.1.2	Ячейка 1.1.3	Ячейка 1.1.4	Ячейка 3.1.1	Ячейка 3.1.2	Ячейка 3.3.1
Ячейка 2.1.1 Ячейка 2.2.1				Ячейка 3.2.1	Ячейка 3.2.2	Ячейка 3.2.3
				Ячейка 4.1.1	Ячейка 4.1.2	
				Ячейка 4.2.1	Ячейка 4.2.2	
				Ячейка 4.3.1	Ячейка 4.3.2	

Табличка

Петровым 2 звонка

Рисунок 9. Образец таблицы

Задание 3. Форматирование таблицы

- Для выполнения этого задания воспользуйтесь результатами выполнения Задания 1.
- В элементе <TITLE> укажите название странички "Задание 3. Форматирование таблицы".
- Сделайте фон первого столбца зеленого цвета, второго - красного, третьего - синего.
- Задайте ширину бордюра таблицы, равным 3 пикселям.
- Ширину таблицы сделайте равной ширине экрана.

- Шрифт, которым написан текст внутри таблицы, сделайте белым.

Задание 4. Заполнение таблицы

- Для выполнения этого задания воспользуйтесь результатами выполнения Задания 1.
- В элементе <TITLE> укажите название странички "Задание 4. Заполнение таблицы".
- В первую ячейку первой строки вставьте рисунок из файла *.jpg.
- Во второй ячейке второй строки напишите ваше любимое четверостишие.
- Сделайте фон третьей ячейки второй строки красного цвета.

Задание 5.

Составьте выписку из своей зачетной книжки, оформив ее в виде таблицы.

Задание 6.

Изучите примеры и создайте WEB-страницу, на которой посередине разместите таблицу следующего вида (ширина таблицы=90%), рисунок 12.

Наиболее популярные модели струйных принтеров				
Изготовитель, название модели	Формат печати	Количество цветов	Макс. разрешение (dpi)	Макс. скорость печати (стр./мин.)
Hewlett Packard				
HP DeskJet 690C+	A4	6	600x600	5
HP DeskJet 970Cxi		4	2400x1200	12
Epson				
Epson Stylus Color 440	A4	4	720x720	4
Epson Stylus Color 900			1440x720	12

Рисунок 12. Образец таблицы

Содержание отчета и его форма

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- ☐ вариант вашего задания;
- ☐ распечатку страниц выполненного задания;
- ☐ выводы по проделанной работе.

Защита работы

Защита работы заключается:

- ☐ в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- ☐ демонстрации преподавателю выполненного задания;
- ☐ в ответах на контрольные вопросы по лабораторной работе;
- ☐ в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

1. Основные методы создания таблиц.
2. Параметры, которые используются при создании таблиц в HTML-документах.
3. Группировка данных
4. Колонтитулы таблицы
5. Прорисовка структуры таблицы
6. Вложенные таблицы
7. Создание цветных таблиц.

Лабораторная работа 7

Тема: Формы в HTML.

Цель работы: познакомиться с разными видами пользовательских форм

Теоретическая часть

Пользовательская форма – это совокупность стандартных HTML-конструкций ввода текстовой и прочей информации и программы-обработчика этой информации, работающей на Web-сервере. Иными словами, пользовательская форма (или HTML-форма) служит для передачи информационных данных серверу.

Результат конструкций языка разметки HTML интерпретируется браузером, с помощью которого пользователь электронного документа получает информацию. Таким образом, объединив все эти формулировки, можно сказать, что HTML-форма выступает в роли посредника между пользователем и сервером.

Посетитель Web-страницы вводит в HTML-форму определенные данные, которые обрабатываются программой и отсылаются на Web-сервер. Все эти действия укладываются в три стадии:

1. Ввод пользователем информации.
2. Обработка введенной информации программой, установленной на сервере.
3. Получение результата отправления введенной информации на Web-сервер (открытие нового HTML-документа, переадресация на предыдущую страницу и пр.).

В качестве программы-обработчика чаще всего выступает CGI-сценарий (скрипт, который обычно разрабатывается на языке Perl или C/C++ и который взаимодействует со специальным компонентом Web-сервера - Common Gateway Interface) или программы, написанные на основе таких серверных языков программирования, как *PHP*, *ASP*, *JSP* и др.

Значение пользовательских форм трудно переоценить – они являются особым средством HTML, дающим посетителю возможность не только пассивно просматривать информацию, но и быть задействованным в содержании Web-

сайта. Такое свойство принято называть интерактивностью, которая на сегодняшний день встречается практически во всех электронных документах.

Диапазон функций, выполняемых HTML-формами, чрезвычайно широк, поэтому остановимся лишь на некоторых, наиболее распространенных направлениях.

Организация обратной связи

Форма обратной связи – это неотъемлемый атрибут Web-сайта любой тематики и функциональной направленности. Размещая электронный документ во Всемирной сети, мы уже подразумеваем реакцию пользовательской аудитории – похвалу и одобрение или же несогласие и критику.

В качестве средства обратной связи между пользователем и автором электронного документа могут выступать следующие.

1. **Почтовая форма.** Посетителя просят указать свои персональные данные и изложить суть обращения. Текст обращения обрабатывается программой и высылается автору сайта по электронной почте. При этом другие посетители сайта не могут получить доступ к содержанию отправленного сообщения;

2. **Гостевая книга.** Электронная книга жалоб и предложений, в которой любой желающий может оставить свой отзыв, поделиться впечатлениями о сайте, изложить критические замечания и указать на ошибки автора. В этом случае сообщения одного посетителя видимы для остальных пользователей гостевой книги;

3. **Интернет-форум.** Более мощный инструмент стимулирования оценки пользовательской аудитории, нежели почтовая форма или гостевая книга. В интернет-форумах происходит многосторонний диалог, причем не только между человеком, задающим вопрос, и автором Web-сайта, но и между всеми пользователями данного интернет-ресурса. На вопрос одного пользователя может ответить другой, третий подскажет то, чего не упомянул второй, наконец, автор сайта подытожит высказывания всех пользователей и даст свой развернутый ответ. Часто в рамках интернет-форума работают дополнительные функцио

нальные средства общения: личные сообщения, скрытые разделы форума для привилегированных пользователей, опросы общественного мнения и пр.;

4. **Чат.** Средство чата (от англ. "to chat" — разговаривать, болтать) подразумевает единовременное общение многих пользователей в режиме реального времени. В отличие от гостевой книги или форума, где ответ на поставленный вопрос может быть выслан и через час, и через день, общение в чате происходит между двумя (или более) собеседниками так, будто они сидят в одной комнате. Собеседники могут находиться в разных городах, государствах и временных зонах, однако, присутствуя в чате, все пользователи способны общаться одновременно. В чатах также могут присутствовать некоторые дополнительные функции: приваты (отдельный разговор между двумя пользователями), система почтовых сообщений, голосования и т. д.

Авторизация

Для получения доступа к информации на многих интернет-ресурсах существует необходимость регистрации в сервисе. После регистрации, которая подразумевает указание уникального пользовательского имени и пароля, пользователь должен пройти процедуру авторизации, т. е. проверки на наличие в базе данных сайта заявленного имени и пароля.

Система авторизации распространена на информационных ресурсах (новостные ленты, афиши событий), корпоративных серверах (закрытые разделы для клиентов, партнеров, дистрибьюторов), сайтах электронной почты и т. д.

Проведение исследований

Для изучения пользовательского спроса на тот или иной вид услуг, выявления и последующего анализа реакции посетителей, на многих сайтах устраиваются всевозможные опросы, анкетирования, голосования, тесты и пр.

Набор стандартных HTML-конструкций облегчает участие пользователя в подобных исследованиях. Все, что ему потребуется, это отметить подходящие варианты ответа и подтвердить участие в опросе.

Пользовательский профиль

Регистрируясь на каком-либо интернет-ресурсе, человек указывает всякого рода информацию: имя, город и страну проживания, увлечения, место работы, контактные данные и пр. Совокупность таких данных составляет пользовательский профиль, который впоследствии возможно будет изменить путем редактирования соответствующих полей HTML-формы.

Структура пользовательских форм

Структура любой пользовательской формы описывается тегом-контейнером `<FORM>`, внутри которого могут располагаться теги и параметры составляющих элементов HTML-формы. В пределах тега `<FORM>` не может содержаться другая пользовательская форма. Закрывающий тег обязателен для указания. Тег `<FORM>` может содержать четыре основных параметра: *ACTION*, *METHOD*, *NAME* и *ENCTYPE*. Рассмотрим каждый из них.

`<FORM>...</form>`

Этот элемент необходим для построения сложных форм. Все формы начинаются тегом `<FORM>` и завершаются тегом `</form>`. После заполнения формы и подтверждения ввода со стороны пользователя введенная информация пересылается на сервер и обрабатывается при помощи CGI-программы, связанной с формой.

Параметр **action**

Параметр **action** является единственным обязательным параметром тега `<FORM>` и предназначен для указания пути на Web-сервере к программе обработчику данных пользовательской формы, например:

```
<form action = "http://www. название. домен/имя программы">
```

Путь к программе-обработчику может быть как относительный, так и абсолютный.

Значение этого параметра играет важную роль с точки зрения работоспособности HTML-формы. Если путь или название программы указаны неверно или же указанный файл не является исполняемым на стороне сервера, то обработка введенных пользователем данных может быть нарушена, произведена некорректно или вовсе не быть осуществленной.

Наиболее распространенными типами исполняемых на стороне сервера файлами являются CGI-программы (с расширениями *pi*, *cgi*, *fcgi*), PHP-скрипты (*php*, *php3*, *phtm*, *phtml*), ASP-приложения (*asp*), JSP-обработчики (*jsp*) и др.

Одним из вариантов обработки формы может быть пересылка данных по электронной почте:

<FORM ACTION ="mailto:Адрес@название.домен">

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задание 1

№	Общие задания
1.	Создайте HTML форму "Личная", в которой пользователь будет задавать свое имя, фамилию, адрес, интересы (может быть несколько), username, password. Предусмотреть текстовое поле для комментария из нескольких строк. Следует учитывать следующие моменты компоновки и дизайна HTML форм выбор даты, месяца и года рождения должен производиться различными способами. Предусмотрите в данной форме гиперссылку для перехода на другой Web документ.
2.	В новом HTML документе разработайте HTML-форму, в которой на примере своей зачетной книжки (дисциплина и оценка), покажите использование различных видов списков для компоновки форм, использование флажков и переключателей.

Содержание отчета и его форма

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- ☐ вариант вашего задания;
- ☐ распечатку страниц выполненного задания;
- ☐ выводы по проделанной работе.

Защита работы

Защита работы заключается:

- ☐ в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- ☐ демонстрации преподавателю выполненного задания;
- ☐ в ответах на контрольные вопросы по лабораторной работе;
- ☐ в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

1. Перечислите основные теги для создания HTML-форм.
2. Тег (дескриптор) FORM.
3. Ввод информации в HTML-форму. Тег INPUT.
4. Создание паролей в HTML-формах.
5. Осуществление выбора информации с помощью флажков и переключателей.
6. Назначение кнопки RESET.
7. Назначение кнопки SUBMIT.
8. Выбор информации в HTML-форме. Основные теги и их характеристика.
9. Характеристика дескриптора TEXTAREA.
10. Какой тег следует указывать при поиске из нескольких массивов информации?
11. Если форма, проектируемая пользователем достаточно велика, то какой способ используют для размещения информации?
12. Какой тег используется для формирования горизонтальной строки?
13. Какой тег используют для перевода строки?
14. Что улучшает совместное использование в Web документе таблиц и форм?
15. Какие элементы управления позволяют передать данные с компьютера клиента на сервер?
16. Напишите пример одновременного использования HTML-форм, таблиц и кнопки передачи запроса.

Лабораторная работа 8

Тема: Технология CSS

Цель работы: познакомиться с технологией CSS

Назначение CSS

Дизайн Web-узлов - это точное размещение компонентов HTML-страниц относительно друг друга в рабочей области окна браузера.

Неточность данного определения Web-дизайна очевидна. В нем не учтены ни цвет, ни форма, ни другие свойства компонентов HTML-страниц. Главное в этом определении - показать ограниченность возможностей HTML-разметки. Позиционирование компонентов на странице является одним из самых слабых мест в HTML.

Компонентами страницы являются: блоки текста, графика и встроенные в страницу приложения. Размер и границы каждого из этих компонентов в рамках HTML-разметки задаются с разной степенью точности. Размер графики и приложений можно задать с точностью до пикселя. Размеры текстовых блоков в HTML задать нельзя. Они вычисляются браузером на основе относительного размера шрифта умолчания.

Автор страницы не может заранее определить настроек браузера пользователя, что существенно ограничивает число вариантов представления информации на странице.

Нельзя сказать, что разработчики браузеров не предпринимали попыток изменить данную ситуацию. В ранних версиях браузеров CERN для платформы NEXT и браузерах WWWC автору страницы давалась возможность переопределять настройки умолчания браузера через HTML-разметку. Но этот подход не получил продолжения в коммерческих продуктах и постепенно завял.

Другой способ управления настройками браузера - программирование на JavaScript. Бурное развитие этого языка заставляет говорить о возможности

полного контроля над процессом отображения HTML-страниц. Недостаток JavaScript - отказ от декларативного характера разметки и относительно большой объем кода для переопределения свойств элементов разметки.

Спецификация CSS (Cascading Style Sheets) позволяет остаться в рамках декларативного характера разметки страницы и дает полный контроль над формой представления элементов HTML-разметки.

Каскадные таблицы стилей призваны разрешить противоречие между точностью определения размеров картинок и приложений с одной стороны, и точностью определения размеров блоков текста и его начертания с другой.

Кроме размера компонентов, таблицы стилей позволяют определить цвет и начертание текстового фрагмента, изменять эти параметры внутри текстового блока, выполнять выравнивание текстового блока относительно других блоков и компонентов страницы.

Наличие всех этих возможностей позволяет говорить о CSS как о средстве разделения логической структуры документа и формы его представления. Логическая структура документа определяется элементами HTML-разметки, в то время как форма представления каждого из этих элементов задается CSS-описателем элемента.

CSS позволяет полностью переопределить форму представления элемента разметки по умолчанию. Например, `<i>...</i>` определяет отображение текста курсивом:

`<i> Отообразим текст курсивом</i>`

Отообразим текст курсивом

А теперь переопределим стиль отображения для элемента разметки `i`:

`<i style="text-decoration:underline;font-style:normal;">`

Отообразим текст курсивом

`</i>`

Этот пример показывает, что привычный стиль отображения элементов может быть полностью изменен при помощи CSS. В данной технологии HTML-разметка носит чисто декларативный характер.

Практическая значимость CSS для Web-инжиниринга (совокупности технологий разработки и сопровождения Web-узлов) заключена в том, что процесс создания узла можно формализовать и представить в виде следующей последовательности действий:

- Сначала нужно определиться с номенклатурой страниц, т.е. все страницы проектируемого Web-узла разбить на типы, например,: домашняя страница, навигационные страницы, информационные страницы, коммуникационные страницы и т.п.. У каждого узла этот перечень может быть своим.

- Для каждого из типов страниц разрабатывается определенная логическая структура (стандартный набор компонентов страницы).

- После этого разрабатывается навигационная карта узла и форма ее реализации на страницах.

- Для каждого стандартного компонента страницы разрабатывается стиль его отображения (CSS-описатель).

- Теперь остается только рисовать картинки, создавать анимацию, писать программы, вручную вводить текст и графику или генерировать содержание страниц автоматически во время обращения к ним.

Объяснив таким образом роль и назначение CSS среди многообразия Web-технологий, мы перейдем непосредственно к обсуждению практики применения каскадных таблиц стилей.

Оборудование и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

Задание 1

Создайте стилевой файл содержащий оформление web-сайта. Обязательно определение оформления следующих элементов:

- Заголовки 1-4 уровней.
- Пункты меню 1-4 уровней.
- Гиперссылки.

- Таблицы и ячейки таблиц.
- Основной текст.

Содержание отчета и его форма

Отчет должен иметь форму согласно оформлению простого реферата. Титульный лист должен включать: название дисциплины, название лабораторной работы, фамилию и инициалы студента, номер группы, фамилию и инициалы преподавателя. Основная часть лабораторной работы должна содержать:

- ☐ вариант вашего задания;
- ☐ распечатку страниц выполненного задания;
- ☐ выводы по проделанной работе.

Защита работы

Защита работы заключается:

- ☐ в выполнении варианта своего задания и оформления отчета в соответствии с подпунктом «Содержание отчета и его форма»;
- ☐ демонстрации преподавателю выполненного задания;
- ☐ в ответах на контрольные вопросы по лабораторной работе;
- ☐ в ответах на дополнительные вопросы по лабораторной работе и дисциплине «Основы работы в Web-среде».

Контрольные вопросы

17. Перечислите основные теги для создания HTML-форм.
18. Тег (дескриптор) FORM.
19. Ввод информации в HTML-форму. Тег INPUT.
20. Создание паролей в HTML-формах.
21. Осуществление выбора информации с помощью флажков и переключателей.
22. Назначение кнопки RESET.
23. Назначение кнопки SUBMIT.
24. Выбор информации в HTML-форме. Основные теги и их характеристика.
25. Характеристика дескриптора TEXTAREA.

26. Какой тег следует указывать при поиске из нескольких массивов информации?

27. Если форма, проектируемая пользователем достаточно велика, то какой способ используют для размещения информации?

28. Какой тег используется для формирования горизонтальной строки?

29. Какой тег используют для перевода строки?

30. Что улучшает совместное использование в Web документе таблиц и форм?

31. Какие элементы управления позволяют передать данные с компьютера клиента на сервер?

32. Напишите пример одновременного использования HTML-форм, таблиц и кнопки передачи запроса.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по организации и проведению самостоятельной работы
по дисциплине «Web-технологии и приложения»
для студентов направления подготовки
11.03.02 Инфокоммуникационные технологии и системы связи

Ставрополь 2026

ОБЩАЯ ХАРАКТЕРИСТИКА ДИСЦИПЛИНЫ

Цель и задачи дисциплины

Целями освоения дисциплины являются формирование у студентов универсальных и общепрофессиональных компетенций, основанных на овладении основами web-технологий, интернет-программирования, изучение принципов обмена данными в сети интернет.

Задачи освоения дисциплины:

1. изучение глобальной сети Интернет, принципов ее построения и способов обмена данными;
2. изучение технологий взаимодействия индивидуального и коллективного пользователя с мировыми информационными ресурсами;
3. изучение технологий создания web-сайта.

Место дисциплины в структуре образовательной программы

Дисциплина «Web-технологии и приложения» относится к дисциплинам обязательной части.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ОПК-3. Способен применять методы поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности	ИД-3 ОПК-3 решает задачи обработки данных с помощью средств вычислительной техники.	решает задачу обработки данных с помощью средств вычислительной техники, используя язык HTML и технологию CSS для создания web-сайтов.

ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ИД-4 ОПК-4 использует возможности вычислительной техники и программного обеспечения для решения задач управления и алгоритмизации процессов обработки информации.	использует возможности вычислительной техники и программного обеспечения для создания web-сайтов, использует язык HTML и технологию CSS для решения задач обработки информации
ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ИД-2 ОПК-5 читает коды программных продуктов, написанных на освоенных языках программирования, и вносит требуемые изменения	читает коды web-сайтов, понимает основные разделы HTML документа и вносит требуемые изменения в HTML документ
	ИД-3 ОПК-5 разрабатывает оригинальные алгоритмы и компьютерные программы, пригодные для практического применения	разрабатывает оригинальные HTML документы, пригодные для практического применения

2. ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ

В основу самостоятельной работы студентов по дисциплине положено конспектирование лекций и рекомендуемых источников, подготовка к компьютерному тестированию знаний, оформление и подготовка к защите отчетов по лабораторным работам, выполнению разноуровневых индивидуальных заданий в ходе лабораторных работ. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, которые студенты предъявляют преподавателю в ходе лабораторных занятий или в часы индивидуальных консультаций.

Таблица 1 - Технологическая карта самостоятельной работы

Код оцениваемой компетенции, индикатора (ов)	Этап формирования компетенции (№ темы) (в соответствии с рабочей программой дисциплины)	Средства и технологии оценки	Вид контроля, аттестация (текущий/промежуточный)	Тип контроля (устный, письменный или с использованием технических средств)	Наименование оценочного средства
ОПК-3 _{ид-3} ОПК-4 _{ид-4} ОПК-5 _{ид-2,3}	Тема 1-9	Вопросы для собеседования по лабораторным работам	текущий	устный	Вопросы для собеседования по лабораторным работам
ОПК-3 _{ид-3} ОПК-4 _{ид-4} ОПК-5 _{ид-2,3}	Тема 1-9	Вопросы для собеседования	текущий	устный	Вопросы для собеседования
ОПК-3 _{ид-3} ОПК-4 _{ид-4} ОПК-5 _{ид-2,3}	Тема 1-9	Вопросы к экзамену	текущий	устный	Вопросы к экзамену

Для успешного освоения дисциплины, необходимо выполнить указанные в таблице виды самостоятельной работы, используя рекомендуемые источники информации.

1. Перечень основной литературы:

1. Крис, Миллз. Введение в HTML5: учебное пособие / Крис Миллз. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 133 с. - Книга находится в базовой версии ЭБС IPRbooks., экземпляров неограничено
2. Савельев, А. О. HTML5. Основы клиентской разработки / А.О. Савельев ; А.А. Алексеев. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 272 с., экземпляров неограничено
3. Серова, Е. А. Использование web-технологий при создании информационных систем Электронный ресурс / Серова Е. А.: учебно-методическое пособие. - Москва : МИСИ – МГСУ, 2020. - 55 с. - ISBN 978-5-7264-2202-2, экземпляров неограничено
4. Ефромеев, Н. М. Основы web-программирования Электронный ресурс : Учебное пособие / Н. М. Ефромеев, Е. В. Ефромеева. - Саратов : Вузовское образование, 2019. - 128 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4487-0529-8, экземпляров неограничено
5. Брокшмидт, К. Введение в разработку приложений для Windows 8 с использованием HTML, CSS и JavaScript / К. Брокшмидт. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 460 с., экземпляров неограничено

2. Перечень дополнительной литературы:

1. Поляков, Е. А. Web-дизайн Электронный ресурс : Учебное пособие / Е. А. Поляков. - Саратов : Вузовское образование, 2019. - 188 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4487-0489-5, экземпляров неограничено

2. Лучанинов, Д.В. Основы разработки web-сайтов образовательного назначения Электронный ресурс : учебное пособие / Д.В. Лучанинов. - Саратов : Ай Пи Эр Медиа, 2018. - 105 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-4486-0174-3, экземпляров неограничено

3. Основы работы в Web-среде: лабораторный практикум : Направление подготовки 11.03.02 Инфокоммуникационные технологии и системы связи. Бакалавриат / сост. С. В. Говорова ; ФГАОУ ВО Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2017. - 160 с., экземпляров неограничено

3. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю)

1. Информационные технологии : лабораторный практикум : Специальности 10.05.03 - Информационная безопасность автоматизированных систем. 11.03.02 - Инфокоммуникационные технологии и системы связи / сост. С. В. Говорова, М. А. Лапина ; Сев.-Кав. федер. ун-т - Ставрополь : СКФУ, 2017(2016). - 168 с.

2. Основы работы в Web-среде : лабораторный практикум : Направление подготовки 11.03.02 Инфокоммуникационные технологии и системы связи. Бакалавриат / сост. С. В. Говорова ; ФГАОУ ВО Сев.-Кав. федер. ун-т - Ставрополь : СКФУ, 2017. - 160 с.

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА

При самостоятельном изучении тем лекционных занятий и заданной литературы студентам рекомендуется:

- в день проведения занятий проработать, изученный на занятии учебный материал по конспекту. При этом необходимо выделить вопросы, на которые следует обратить большее внимание при дальнейшем изучении текущей темы. После проработки учебного материала необходимо ответить на, рекомендуемые контрольные вопросы;
- с целью качественного закрепления изученного материала в последующем следует более детально проработать учебный материал с использованием рекомендованной литературы. Студентам рекомендуется выделить вопросы, требующие более детальной проработки с помощью преподавателя;
- накануне проведения лекции студенту рекомендуется закрепить ранее изученный теоретический материал, подготовить вопросы, требующие уточнения у преподавателя.

Рекомендации по организации работы с литературой

Для овладения, закрепления и систематизации знаний студентам рекомендуется самостоятельная работа с рекомендованной литературой и конспектом лекций. При самостоятельной работе с рекомендованной литературой студентам рекомендуется проводить конспектирование учебного материала, изложенного в рекомендованных источниках.

Конспектирование источников проводится при детальном изучении темы, при этом студентам рекомендуется:

- дополнять конспект лекционного занятия путем его расширения по наиболее важным вопросам, рассмотренным на занятии.

Контроль этого вида самостоятельной работы осуществляется на учебных занятиях путем проверки преподавателем конспекта лекций и собеседования.

При конспектировании источников студенты должны исходить из рационального объема, конспектируемого материала. Конспект должен быть кратким и понятным при его использовании после изучения текущей темы.

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации.

Работа по конспектированию источников оценивается удовлетворительно если:

- конспект имеет достаточный объем детализации по, изучаемой теме, обеспечивающий заданный уровень освоения теоретического материала дисциплины;
- конспект оформлен аккуратно, изучаемый материал изложен последовательно в соответствии с порядком изучения дисциплины, содержит обобщающие выводы по каждой теме.

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

При выполнении практикума по дисциплине (лабораторные работы) для достижения базового уровня сформированности компетенций студент должен уметь применять базовые знания по разделам дисциплины, владеть навыками самостоятельного решения учебных задач базового уровня.

При выполнении практикума по дисциплине (лабораторные работы) для достижения повышенного уровня сформированности компетенций студент должен уметь применять повышенные знания по разделам дисциплины, владеть навыками самостоятельного решения учебных задач повышенного уровня.

Методические указания при подготовке к лабораторной работе

Подготовку к лабораторной работе рекомендуется проводить в следующей последовательности:

- уяснить тему и цель, предстоящей лабораторной работы;

- изучить теоретический материал в соответствии с темой лабораторной работы (рекомендуется использовать рекомендованную литературу, конспект лекций, учебное пособие (практикум по лабораторным работам);
- ознакомиться с оборудованием и материалами, используемыми на лабораторной работе (при использовании специализированного оборудования или программного обеспечения необходимо изучить порядок и правила его использования).

Методические указания при выполнении лабораторной работы

При выполнении лабораторной работы студенты должны строго соблюдать, установленные требования безопасности.

При выполнении лабораторной работы студентам рекомендуется:

- уяснить цель, выполняемых заданий и способы их решения;
- задания, указанные в лабораторной работе выполнять в той последовательности, в которой они указаны в лабораторном практикуме;
- при выполнении практического задания и изучении теоретического материала использовать помощь преподавателя;
- оформить отчет по лабораторной работе;
- ответить на контрольные вопросы.

Методические указания при подготовке к защите лабораторной работы

При подготовке к защите лабораторной работы студентам рекомендуется:

- подготовить отчет по лабораторной работе;
- подготовить обоснование, сделанных выводов;
- закрепить знания теоретического материала по теме лабораторной работы (рекомендуется использовать контрольные вопросы);
- знать порядок проведения расчетов (проводимых исследований);
- уметь показать и пояснить порядок исследований при использовании специализированного оборудования или программного обеспечения.

Защита лабораторной работы осуществляется путем собеседования студента с преподавателем. При собеседовании студент представляет на проверку отчет по лабораторной работе. Собеседование со студентом, претендующим на оценку «отлично» проводится по вопросам и практическим заданиям повышенного уровня обучения. Собеседование со студентом, не претендующим на оценку «отлично» проводится по вопросам и практическим заданиям базового уровня обучения.

5. ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ ПО САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ ЛИТЕРАТУРЫ

Вопросы (задача, задание) для проверки уровня обученности

1. История возникновения и развития глобальной компьютерной сети Интернет.
2. Основные принципы работы Интернет
3. Основные принципы доступа к Интернет.

4. История создания языка разметки текста.
5. Основные разделы HTML документа
6. Понятие тэга. Основные тэги HTML.
7. Оформление текста в HTML
8. Гипертекстовые ссылки. Структура ссылок в HTML-документе
9. Ссылки на точки внутри документа
10. Графика внутри HTML-документа. Карты в HTML.
11. Списки в HTML
12. Средства описания таблиц в HTML. Создание таблиц в HTML
13. Задание формы — элемент FORM
14. Определение элементов управления формы — тег INPUT. Атрибуты тега INPUT
15. Технология CSS. Подключение таблиц стилей
16. Каскадность стилей. Синтаксис CSS
17. Псевдоклассы
18. Анатомия CSS набора правил
19. Разные типы селекторов (селектором тега, ID селектор, Селектор класса, Селектор атрибута, Селектор псевдокласса)
20. История создания JavaScript, основные понятия и определения. Способы использования JavaScript внутри HTML-документа
21. Использование списков в форме — тег SELECT. Атрибуты тега SELECT
22. Фреймы в HTML.
23. Условные операторы JavaScript
24. Операторы цикла (Оператор for, Оператор for-in, Оператор while)
25. Оператор break .Оператор continue в JavaScript
26. События JavaScript
27. Объект Window в JavaScript
28. Основные методы объекта document в JavaScript
29. Методы объекта History
30. Объект navigator
31. Объект anchor и массив anchors
32. Объект image и массив images
33. Объект link и массив links
34. Массив elements
35. Объект form и массив forms
36. Использование массива forms
37. Основные методы объекта document.
38. Типы сайтов. Принципы web-дизайна

1. Критерии оценивания компетенций (в соответствии с результатами освоения дисциплины)

Оценка «отлично» выставляется студенту, если он четко решает задачу, используя язык HTML и технологию CSS для создания web-сайтов. Полно использует язык HTML и технологию CSS для решения задач обработки

информации. Грамотно читает коды web-сайтов, понимает основные разделы HTML документа и вносит требуемые изменения в HTML документ.

Оценка **«хорошо»** выставляется студенту, если он достаточно полно решает задачу обработки данных, используя язык HTML и технологию CSS для создания web-сайтов. В основном правильно использует возможности программного обеспечения для создания web-сайтов, использует язык HTML и технологию CSS для решения задач обработки информации. Читает коды web-сайтов, понимает основные разделы HTML документа и вносит требуемые изменения в HTML документ, допуская незначительные ошибки. В основном точно разрабатывает оригинальные HTML документы, пригодные для практического применения. Может допускать незначительные ошибки.

Оценка **«удовлетворительно»** выставляется студенту, если он слабо решает задачу обработки данных, используя язык HTML и технологию CSS для создания web-сайтов. Слабо использует возможности программного обеспечения для создания web-сайтов, использует язык HTML и технологию CSS для решения задач обработки информации. Читает коды web-сайтов, понимает основные разделы HTML документа и вносит требуемые изменения в HTML документ, допуская значительные ошибки. Допускает серьезные ошибки при разработке оригинальных HTML документов, пригодных для практического применения.

Оценка **«неудовлетворительно»** выставляется студенту, если он не решает задачу обработки данных, используя язык HTML и технологию CSS для создания web-сайтов. Не использует возможности программного обеспечения для создания web-сайтов, не использует язык HTML и технологию CSS для решения задач обработки информации. Не может читать коды web-сайтов, не понимает основные разделы HTML документа и не вносит требуемые изменения в HTML документ. С трудом разрабатывает оригинальные HTML документы, пригодные для практического применения.

2. Описание шкалы оценивания

Успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

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

Процедура проведения данного оценочного мероприятия включает в себя:

- отдельное собеседование по каждой из указанных тем;
- составление конспекта.

Предлагаемые задания позволяют проверить компетенции ПК-2, ПК-3

Для подготовки к каждой теме следует изучить и законспектировать рекомендованные источники и научную литературу и на их основании подготовиться к указанным в методических материалах вопросам для обсуждения.

При проверке задания, оцениваются

- уверенное владение материалом;
- умение четко и логично излагать свои мысли;
- умение творчески подходить к решению основных вопросов темы;

- самостоятельность мышления,
- полное соответствие конспекта установленным требованиям;
- самостоятельная устная речь, полностью раскрывающая суть суть проблематики собеседования.

7.ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ ПРИ ПРОВЕРКЕ ГОТОВНОСТИ К ЛАБОРАТОРНЫМ РАБОТАМ

Тема: Основы языка HTML.

1. Суть и составные части Web технологии.
2. Основы языка разметки гипертекста - HTML
3. Тэг <HTML>
4. Элемент <BODY>
5. Заголовочные тэги
6. META - метаинформация
7. TITLE - "внешний" заголовок (титул)
8. Комментарии
9. Оформление текста
- 10.Списки

Тема: Гиперссылки. Внедрение изображений

1. Гипертекстовые ссылки
2. Структура ссылок в HTML-документе
3. Ссылки на точки внутри документа
4. Графика внутри HTML-документа
- 5.
6. Тема: Оформление HTML документа. Таблицы
7. Средства описания таблиц в HTML
8. Создание таблиц в HTML

Тема: Формы в HTML

1. Задание формы — элемент FORM
2. Определение элементов управления формы — тег <INPUT>
3. Атрибуты тега <INPUT>
4. Использование списков в форме — тег <SELECT>
5. Атрибуты тега <SELECT>

Тема: Технология CSS. Проектирование оформления web-сайта

1. Технология CSS
2. Подключение таблиц стилей
3. Описание в секции заголовка
4. Вынесение описания стилей во внешний файл
5. Каскадность стилей
6. Синтаксис CSS

7. Псевдоклассы
8. Основные параметры CSS
9. Примеры описания таблицы стилей

Тема: Проектирование структуры web-сайта

1. Типы сайтов
2. Принципы web-дизайна

Тема: Внедрение объектов JavaScript в HTML-документ

1. Способы использования JavaScript внутри HTML-документа
2. Встраивание в тело документа HTML при помощи тега `<SCRIPT>` и `</SCRIPT>`
3. Ссылки на файлы с подгружаемыми скриптами
4. Встраивание с применением функций

Тема: Базовые конструкции языка JavaScript

1. Условные операторы
2. Операторы цикла
3. Оператор for
4. Оператор for-in
5. Оператор while
6. Оператор break
7. Оператор continue

1. Критерии оценивания компетенций

оценка «отлично» выставляется, если студент продемонстрировал высокое умение применять полученные знания на практике через решение конкретной задачи по применению web-технологий, свободно без затруднений справился с поставленной задачей, показав владение разносторонними приемами и навыками ее выполнения, не допустил ошибок и неточностей;

оценка «хорошо» выставляется, если студент продемонстрировал умение применять полученные знания на практике через решение конкретной задачи по применению web-технологий, справился с поставленной задачей, показав владение необходимыми приемами и навыками ее выполнения, при этом допустил не более одной ошибки;

оценка «удовлетворительно» выставляется, если студент продемонстрировал посредственное умение применять полученные знания на практике через решение конкретной задачи по применению web-технологий, с трудом справился с поставленной задачей, при этом допустил не более двух ошибок;

оценка «неудовлетворительно» выставляется, если студент не продемонстрировал умение применять полученные знания на практике через решение конкретной задачи по применению web-технологий, не справился с

поставленной задачей или допустил при ее решении три и более серьезные ошибки.

2. Описание шкалы оценивания

Успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

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

Процедура проведения данного оценочного мероприятия включает в себя:

- собеседование;
- защита лабораторных работ.

Предлагаемые задания позволяют проверить компетенции ПК-2, ПК-3.

К лабораторному занятию студент должен подготовить ответы на вопросы, проработать дополнительные научные источники, сделать конспект теоретического материала (если это предусмотрено).

Максимальное количество баллов студент получает, если он выполнит лабораторное задание, владеет материалом, умеет логично и четко излагать мысли, показывает самостоятельность выполнения задания.

Основанием для снижением оценки являются:

- слабое знание темы и неполностью выполненное задание;
- невыполненное задание к лабораторной работе;
- отсутствие умения применить теоретические знания для решения лабораторных задач.

При проверке задания, оцениваются

- активное участие в работе;
- уверенное владение материалом;
- умение четко и логично излагать свои мысли;
- умение творчески подходить к решению заданий;
- самостоятельность мышления,
- самостоятельная устная речь, полностью раскрывающая суть сути проблематики собеседования.

8.ВОПРОСЫ К ЭКЗАМЕНУ

Вопросы (задача, задание) для проверки уровня обученности

1. История возникновения и развития глобальной компьютерной сети Интернет.
2. Основные принципы работы Интернет.
3. Основные принципы доступа к Интернет.
4. Адресация.
5. Стек протоколов Интернет.
6. Серверы имен. Понятие зоны, деление DNS-пространства на зоны.
7. История создания языка разметки текста.
8. Основные разделы HTML документа.
9. Понятие тэга. Основные тэги HTML.

10. Гипертекстовые ссылки
11. Структура ссылок в HTML-документе
12. Ссылки на точки внутри документа
13. Графика внутри HTML-документа
14. Средства описания таблиц в HTML
15. Создание таблиц в HTML
16. Задание формы — элемент FORM
17. Определение элементов управления формы — тег <INPUT>
18. Атрибуты тега <INPUT>
19. Использование списков в форме — тег <SELECT>
20. Атрибуты тега <SELECT>
21. Технология CSS
22. Подключение таблиц стилей
23. Описание в секции заголовка
24. Вынесение описания стилей во внешний файл
25. Каскадность стилей
26. Синтаксис CSS
27. Псевдоклассы
28. Основные параметры CSS
29. Примеры описания таблицы стилей
30. Типы сайтов
31. Принципы web-дизайна
32. История создания JavaScript, основные понятия и определения.
33. Способы использования JavaScript внутри HTML-документа (Встраивание в тело документа HTML при помощи тега <SCRIPT> и </SCRIPT>. Ссылки на файлы с подгружаемыми скриптами. Встраивание с применением функций)
34. Условные операторы
35. Операторы цикла (Оператор for, Оператор for-in, Оператор while)
36. Оператор break .Оператор continue
37. События JavaScript
38. Классы и объекты JavaScript
39. Объект Window
40. Основные методы объекта document
41. Методы объекта History
42. Объект navigator
43. Объект anchor и массив anchors
44. Объект image и массив images
45. Объект link и массив links
46. Массив elements
47. Объект form и массив forms
48. Использование массива forms
49. Объекты image и массив images
50. Загрузка новых изображений.

1. Критерии оценивания компетенций (в соответствии с результатами освоения дисциплины)

Оценка «отлично» выставляется студенту, если студент дал полный, развернутый ответ на поставленные вопросы, при этом показана совокупность осознанных знаний по дисциплине, доказательно раскрыты основные положения вопросов; в ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий. Знание демонстрируется на фоне понимания его в системе дисциплины. Ответ изложен литературным языком с использованием технической терминологии. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа, продемонстрировал высокое умение применять полученные знания на практике через решение конкретной задачи, свободно без затруднений справился с поставленной задачей, показав владение разносторонними приемами и навыками ее выполнения, не допустил ошибок и неточностей.

Оценка «хорошо» выставляется студенту, если студент дал полный, развернутый ответ на поставленные вопросы, при этом показано умение выделить существенные и несущественные признаки, причинно-следственные связи. Ответ четко структурирован, логичен, изложен литературным языком с использованием технической терминологии. Могут быть допущены 2-3 неточности или незначительные ошибки, исправленные студентом с помощью преподавателя, продемонстрировал умение применять полученные знания на практике через решение конкретной задачи, справился с поставленной задачей, показав владение необходимыми приемами и навыками ее выполнения, при этом допустил не более одной ошибки.

Оценка «удовлетворительно» выставляется студенту, если студент дал недостаточно полные и недостаточно развернутые ответы по вопросам билета. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. В ответе отсутствуют выводы. Умение раскрыть значение обобщенных знаний не показано. Речевое оформление требует поправок, коррекции, продемонстрировал посредственное умение применять полученные знания на практике через решение конкретной задачи, с трудом справился с поставленной задачей, при этом допустил не более двух ошибок.

Оценка «неудовлетворительно» выставляется студенту, если ответ представляет собой разрозненные знания с существенными ошибками по вопросу. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь обсуждаемых вопросов с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная, гистологическая терминология не используется. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента, ответ на

вопросы полностью отсутствует или студент отказался от ответа, не продемонстрировал умение применять полученные знания на практике через решение конкретной задачи, не справился с поставленной задачей или допустил при ее решении три и более серьезные ошибки.

2. Описание шкалы оценивания

Успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

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

Процедура проведения экзамена осуществляется в соответствии с Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры - в СКФУ.

В экзаменационный билет включаются два вопроса.

Для подготовки по билету отводится 20 минут.

При подготовке к ответу студенту предоставляется право пользования рабочей программой.

При проверке практического задания/задачи, оцениваются логичность, точность и аргументированность решения