**Министерство образования Республики Башкортостан**

**УФИМСКИЙ КОЛЛЕДЖ РАДИОЭЛЕКТРОНИКИ, ТЕЛЕКОММУНИКАЦИЙ И БЕЗОПАСНОСТИ**

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Зам. директора  \_\_\_\_\_\_\_\_\_\_\_\_\_ Л.Р. Туктарова  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2015 г. |

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛЕНИЮ ПРАКТИЧЕСКИХ (ЛАБОРАТОРНЫХ) РАБОТ И контрольнЫХ ЗАДАНИЙ**

**ДЛЯ СТУДЕНТОВ ЗАОЧНОЙ ФОРМЫ ОБУЧЕНИЯ**

**ПО ДИСЦИПЛИНЕ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»**

**(раздел WEB-технологии)**

***специальность 09.02.01 «Компьютерные системы и комплексы»***

|  |  |
| --- | --- |
|  | РАЗРАБОТЧИК  \_\_\_\_\_\_\_\_\_\_\_ Р.Р. Рахимов  РАССМОТРЕНО  на заседании кафедры «Программирование и информационные технологии»  \_\_\_\_\_\_\_\_\_\_\_\_\_ М.Е. Бронштейн  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2015 г. |

**Уфа 2015 г.**

**Порядок выполнения отчета по лабораторной (практической) работе**

1. Ознакомиться с теоретическим материалом по практической работе.
2. Записать краткий конспект теоретической части.
3. Выполнить предложенное задание согласно варианту по списку группы.
4. Продемонстрировать результаты выполнения предложенных заданий преподавателю.
5. Записать код программы в отчет.
6. Ответить на контрольные вопросы.
7. Записать выводы о проделанной работе.

**Практическая работа № 1**

**«Разметка и эскиз web-страницы тегами HTML»**

**Цель работы:** изучение на практике теоретических аспектов web-дизайна. Создание первой web-страницы.

**Образовательные результаты, заявленные во ФГОС третьего поколения:**

Студент должен

уметь:

- составлять HTML-код страницы;

- проектировать web-приложения;

знать:

- синтаксис HTML.

**Краткие теоретические и учебно-методические материалы по теме практической работы**

Всевозможных реализаций языка разметки гипертекста существует довольно много, но для изучения лучше всего воспользоваться HTML версии 3.2.

В среде web-дизайнеров почему-то принято считать, что на свете существуют всего две методики создания web-страниц. Первая заключается в использовании объектно-ориентированных сред разработки html-документов, так называемых web-редакторов. Web-мастеру, использующему в своей работе подобную программу, по большому счету не нужно знать HTML: достаточно лишь уметь обращаться с данным приложением и щелкать мышью. Весь необходимый код редактор генерирует сам. Второй подход, считающийся «верхом профессионализма», подразумевает написание всего кода вручную, с применением стандартного текстового редактора MS Windows —программы Notepad, известной также как Блокнот.

**HTML**

**1. Общая структура**

Как и любой другой язык программирования, HTML подразумевает некую стандартизированную структуру построения программы — в данном случае, **html-документа**. Такая структура описывает очередность следования ряда обязательных блоков, которые содержат непосредственно программный код. **Директивы HTML** **называются** **«тегами»** (от англ, tag — отметка). **Теги HTML** заключаются в угловые скобки, синтаксис их записи в общем виде выглядит как **<тег>**. Все объекты, не заключенные в угловые скобки, интерпретатор воспринимает как текстовые элементы, отображая их на экране компьютера «как есть». Итак, упрощенная структура документа HTML выглядит следующим образом (рис. 1.1).

**Виды тегов:**

- *парные,* охватывающие какой-то фрагмент текста и/или другие теги,

***<парный-тег>***текст или другие теги***</парный-тег>***

Парные теги должны вкладываться друг в друга без пересе­чений, т. е. если в области действия тега А открылся тег В, он должен закрыться до того, как закроется тег А.

***<A> <B> </B> </A>***

- *непарные,* стоящие в одиночестве*:*

***<непарный-тег>***

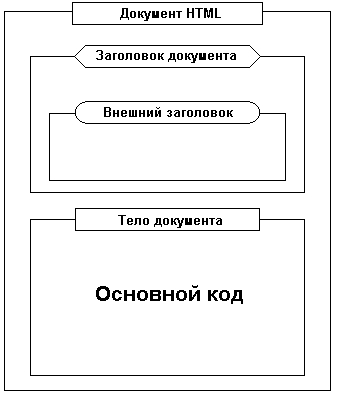


Рис. 1.1. Упрощенная структура html-документа

Как видно из иллюстрации, основной, глобальной конструкцией внутреннего кода web-страницы является нечто, обозначенное на рисунке прямоугольником с надписью **«Документ HTML»**. Это — специальная команда, призванная «объяснить» броузеру, что он имеет дело именно с документом HTML, а не с текстовым или другим файлом. Эта команда называется **«тегом верхнего уровня»** и записывается:

**< НТМL >Содержимое </НТМL>**

Тег верхнего уровня — парный, и его содержимое и есть весь код HTML, составляющий документ.

Следующий на рисунке прямоугольник, обозначенный на схеме как **«Заголовок документа»**. Заголовок web-страницы содержит исчерпывающую информацию о самом документе, а иногда также специальные директивы транслятора, подсказывающие встроенному в броузер интерпретатору HTML правила, по которым следует обрабатывать составляющий страницу код. **Содержимое заголовка не отображается в броузере** и не влияет на внешний вид документа: это служебная информация, которая необходима, прежде всего, самому броузеру. Синтаксис тега заголовка в общем виде выглядит так:

**<HEAD> Содержимое </HEAD>**

Из приведенной схемы видно, что раздел **HEAD** следует в html-документе непосредственно за тегом <HTML> и **является второй обязательной командой**, которую необходимо включать в код web-страницы.

**«Внешний заголовок»**, который, как показано на схеме, является вложенной командой тега <HEAD>. Мнемоника внешнего заголовка записывается следующим образом:

**< TITLE >Внешний заголовок</TITLE>**

**«Внешний заголовок»** отличается от просто «заголовка» документа HTML тем, что именно он отображается в верхнем поле броузера в качестве названия страницы при ее открытии, и именно значение тега **<TITLE>** подставляется по умолчанию в соответствующее диалоговое окно, когда пользователь заносит документ в папку «избранное» (рис. 1.2).

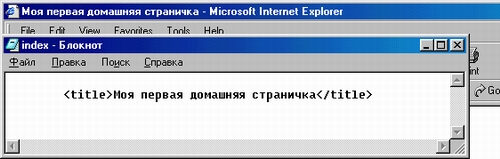


Рис. 1.2. Значение тега <TITLE>

Последняя структурная составляющая кода web-страницы — раздел **«Тело документа»**.

Тело документа, описываемое тегами **<BODY> </BODY>**, включает в себя весь основной код разметки страницы, который и определяет отображение html-документа на экране монитора. Основной текст, иллюстрации, элементы навигации и все, что вы хотите продемонстрировать посетителям вашего сайта, размещается внутри данного тега.

Теперь вы располагаете достаточным объемом информации для того, чтобы представить абстрактную графическую схему (рис. 1.1) в виде конкретного кода HTML:

**<HTML>**

**<HEAD>**

**<TITLE>Моя первая web-страничка</TITLE>**

**</HEAD>**

**<BODY>**

**</BOOY>**

**</HTML>**

Теперь необходимо проделать следующие действия: открыть Блокнот (**Пуск > Все программы > Стандартные > Блокнот**), набрать в нем полученный выше код, сохранить текущий файл в папке **«Site\_vasha\_familiya»** под именем **index.html**, воспользовавшись функцией «Сохранить как», и открыть его в броузере путем двойного щелчка мышью на значке данного файла. Вы получили свою первую web-страничку (рис. 1.3).

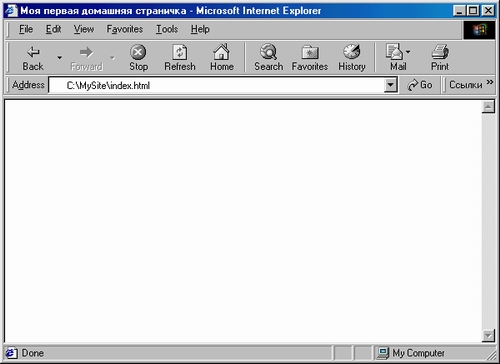


Рис. 1.3. Первая web-страничка

**Этот html-документ отображается в главном окне броузера абсолютно пустым.** Потому что в него не внесено никакого контекстного содержимого. Для того чтобы это содержимое наполнило вашу страничку, необходимо изучить базовые теги разметки гипертекста. А для начала я предлагаю обратить пристальное внимание на приведенную ниже информацию, которая может пригодиться вам в последующей работе.

**2. Атрибуты**

**Тег** — это некая команда HTML, указывающая интерпретатору броузера, каким образом он должен обрабатывать соответствующее каждой конкретной директиве значение. Это значение называется **атрибутом** тега.

Тег может иметь атрибут или не иметь его. Например, тег верхнего уровня <HTML> не имеет атрибутов, а в строке создания новой таблицы <TABLE BORDER="1"WIDTH="100"> выражение «BORDER="1" WIDTH="100"» является атрибутом тега <TABLE>.

В общем виде синтаксис записи тега HTML в совокупности с его атрибутами выглядит следующим образом:

**<ТЕГ ИМЯ\_АТРИБУТА-1="ЗНАЧЕНИЕ" ИМЯ\_АТРИБУТА-2="ЗНАЧЕНИЕ" ...ИМЯ\_АТРИБУТА-n="ЗНАЧЕНИЕ">**

В данном примере первый атрибут тега <TABLE> — это параметр "BORDER", второй атрибут — параметр "WIDTH", а их значения составляют соответственно "1" и "100". Значения атрибутов заключаются в прямые кавычки, записываемые символом «"». Если внутри атрибута какого-либо тега встречается другое значение, заключенное в кавычки, то есть имеет место вложение одних кавычек в другие, в качестве «внутренних» кавычек рекомендуется использовать одинарные, записываемые символом «'». В общем случае такое выражение выглядит следующим образом:

**<ТЕГ ИМЯ\_АТРИБУТА-1="ЗНАЧЕНИЕ-1; 'ЗНАЧЕНИЕ-2'; ЗНАЧЕНИЕ-3">**

Спецификация языка HTML позволяет опускать кавычки для следующих видов атрибутов:

* Атрибуты, записываемые только строчными или заглавными символами латинского алфавита и не включающие иных символов, например, цифр.
* Атрибуты, состоящие только из цифр от 0 до 9.
* Атрибуты, обозначающие промежутки времени.

Например, запись атрибутов BORDER="1" или ALIGN="CENTER" допустимо представить как BORDER=1 и ALIGN=CENTER.

**3. Регистр записи команд**

HTML является одним из немногих языков программирования, не чувствительных к регистру, в котором вы записываете команды (в данном случае — теги и имена их атрибутов). Интерпретатор броузера одинаково обработает директивы <HEAD>, <head>, <Head> или <hEaD>. Таким образом можно записывать теги как в строчном, так и в заглавном варианте. Однако рекомендуется все же применять для написания тегов HTML заглавный регистр, прежде всего, для улучшения читаемости кода, поскольку в этом случае команды языка будет легко отличить от интегрированного в документ информационного текста. Единственным исключением из этого правила служат два элемента HTML, небезразличных к регистру. Это адреса URL и escape-последовательности, о которых мы подробно побеседуем ниже.

**4. Escape-последовательности (подстановки)**

Запись escape-последовательности подразумевает замену служебных символов специальными командами, которые в процессе интерпретации документа замещаются выводимыми на экран искомыми знаками. Все escape-последовательности начинаются с символа амперсанд (**«&»**) и заканчиваются точкой с запятой, а между ними размещается сама команда, записываемая в строчном регистре. В табл. 1.1 приведен список наиболее часто используемых escape-последовательностей.

Таблица 1.1. Наиболее распространенные escape-последовательности

|  |  |  |
| --- | --- | --- |
| **Символ** | **Значение** | **Команда** |
| < | Символ «меньше, чем»; левая угловая скобка | &lt; |
| > | Символ «больше, чем»; правая угловая скобка | &gt; |
| " | Прямые кавычки | &quot; |
| & | Амперсанд | &атр; |
|  | Неразрывный пробел (словосочетания, разделенные таким пробелом, не разрываются при переносе) | &nbsp; |
| © | Символ копирайта | &сору; |
| ® | Символ зарегистрированной торговой марки | &reg; |

Escape-последовательности записываются в тексте по тем же правилам, по которым употребляются обозначаемые ими символы, иными словами, нотация символьного объекта просто вставляется в тот участок текста, где по расчетам web-мастера должен следовать соответствующий служебный символ.

**5. Перевод строки, пробелы, табуляция**

При подготовке web-страниц обязательно следует помнить, что введенное пользователем с помощью клавиатуры форматирование текста, включающее в себя пробелы, отступы табуляции и переводы строки, **игнорируются броузером** при интерпретации html-документа.

Иными словами, такие элементы текста, как

**данное   словосочетание**

или

**данное**

**словосочетание**

на экране монитора будут выглядеть одинаковым образом:

**данное словосочетание**

Таким образом, любое количество пробелов между словами, отступ, выполненный посредством клавиши «Tab», или перевод строки при нажатии клавиши «Enter» преобразуются при компиляции документа в один символ пробела. Поэтому для форматирования основного текста на web-странице применяются специальные теги. Исключением из этого правила может служить только текст, помеченный специальным тегом **<PRE>** — командой предварительного форматирования. Мнемоника записи этого тега выглядит следующим образом:

**<PRE>**

предварительно отформатированный текст

**</PRE>**

В этом случае все, что расположено между открывающим и закрывающим тегами <PRE>, а в нашем примере это выражение «предварительно отформатированный текст», отобразится в окне броузера с сохранением всех введенных дизайнером пробелов, переводов строк и отступов табуляции. Однако злоупотреблять применением данной команды не рекомендуется, поскольку в силу несхожести алгоритмов обработки кода HTML различными броузерами возможно искажение отформатированного таким образом текста в зависимости от экранных настроек пользователей. Помните, что **внутри тега PRE запрещено использовать другие теги форматирования текста.**

**6. Комментарии**

Код HTML может содержать **комментарии**, предназначенные как для самого web-мастера, создающего документ, так и для постороннего пользователя, читающего листинг чужой странички. И в том и в другом случае комментарии записывают с целью избежать возможной путаницы при изучении кода html-документа. **Комментарии не обрабатываются интерпретаторами броузеров** **и не отображаются на экране**, они остаются в составе кода web-страницы «как есть». Комментарий открывается символьной последовательностью, начинающейся с открывающей угловой скобки, восклицательного знака и двух знаков тире: **<!--**, закрывается же несколько иным сочетанием символов: -->. В общем случае синтаксис записи комментария будет выглядеть следующим образом:

**<!-текст комментария -->**

Иногда в комментарий включается информация о создателе документа, предупреждение об авторских правах или адрес электронной почты администратора сайта. Однако чаще всего данный элемент кода HTML используется по своему прямому назначению, для пояснения назначения тех или иных участков листинга html-документа.

**7. Общие рекомендации**

Создатель языка HTML, консорциум W3C, рекомендует включать в качестве первой строки html-кода перед тегом <HTML> так называемый **дефинитивный определитель**, показывающий, что данная web-страница создана именно с использованием HTML версии 3.2, а не какой-либо другой. Определитель не несет явной функциональной нагрузки и не влияет на отображение документа в броузере, однако спецификации требуют его использования. Синтаксис записи данного определителя всегда одинаков:

**<! DOCTYPE HTML PUBLIC "-//W3C//DTO HTML 3.2//EN">**

**8. Цветовые спецификации**

Некоторые элементы языка HTML могут иметь в качестве атрибутов переменные, обозначающие цвета. Эти переменные применяются в основном для определения цвета фона html-документа, цвета текста или цвета, при помощи которого отображаются гиперссылки.

**Определить тот или иной цвет** при создании web-страницы можно двумя методами:

1. посредством задания специальных цветовых меток, обозначающих названия цветов словами;
2. с помощью шестнадцатеричного цифрового кода, заменяющего эти метки.

Известно, что для вывода цветных изображений на экран монитора применяется так называемая **палитра RGB**. При помощи цветовых меток или цифрового кода web-мастер имеет возможность использовать в документе HTML любой цвет, который поддается отображению в рамках стандарта RGB.

Цветовые нотации, заданные посредством символьных меток, легче запоминаются и интуитивно понятны любому программисту, хотя бы немного знакомому с английским языком.

Существует **140 цветов**, для которых приняты стандартные символьные метки.

Цветов же, которые возможно отобразить посредством цифрового кода, несоизмеримо больше.

**Не все броузеры способны отображать некоторые цвета, заданные символьной нотацией**. Например, Netscape Navigator распознает их значительно больше, нежели Microsoft Internet Explorer.

Для обозначения цвета шестнадцатеричным цифровым кодом принят следующий синтаксис записи: перед самим кодом ставится символ **«#»**, далее следует набор из шести знаков в шестнадцатеричной системе счисления.

Практике активно используется шестнадцать цветов. Именно они были приняты в качестве стандарта для шестнадцатицветовой палитры VGA. Обозначение данных цветов приведено в табл. 1.2.

Таблица 1.2. Стандартные цвета

|  |  |  |  |
| --- | --- | --- | --- |
| **Цвет** | **Значение RGB** | **Символьная метка** | **Цифровой код** |
| Белый | 255 255 255 | White | #FFFFFF |
| Черный | 000 | Black | #000000 |
| Зеленый | 0 128 0 | Green | #008000 |
| Светло-зеленый | 0 255 0 | Lime | #00FF00 |
| Серый | 128 128 128 | Gray | #808080 |
| Светло-серый | 192 192 192 | Silver | #C0C0C0 |
| Желтый | 255 255 0 | Yellow | #FFFF00 |
| Темно-бордовый | 128 0 0 | Maroon | #800000 |
| Синий | 0 0 255 | Blue | #0000FF |
| Темно-синий | 0 0 128 | Navy | #000080 |
| Голубой | 0 255 255 | Aqua | #00FFFF |
| Изумрудный | 0 128 128 | Teal | #008080 |
| Красный | 255 0 0 | Red | #FF0000 |
| Пурпурный | 128 0 128 | Purple | #800080 |
| Розовый | 255 0 255 | Fuchsia | #FF00FF |
| Оливковый | 128 128 0 | Olive | #808000 |

Помните, что для передачи цветов в html-документе предпочтительнее все же пользоваться **шестнадцатеричным цифровым кодом**, только так вы можете быть уверены, что броузер пользователя распознает требуемый цвет корректно.

**9. Базовые теги разметки гипертекста**

Рассмотрим основные директивы HTML, которые позволяют поместить в создаваемый документ какое-либо содержательное наполнение.

Существовать специальные команды, позволяющие отформатировать текст на экране именно так, как это необходимо web-мастеру. Подобные команды существуют, и первая из них — это **тег обозначения стандартного абзаца**.

Для того чтобы вывести в составе html-документа **текстовый абзац**, необходимо заключить соответствующий отрывок текста в теги **<Р>** и **</Р>.** Синтаксис записи такой команды выглядит следующим образом:

**<P ALIGN=APГУMEHT>**

**Текст, заключаемый в абзац**

**</P>**

Аргументом атрибута **«ALIGN»** может служить одно из четырех логических выражений, значение которых приведено ниже:

**RIGHT** — выравнивание текста по правому краю экрана или столбца таблицы;

**LEFT** — выравнивание текста по левому краю экрана или столбца таблицы;

**CENTER** — выравнивание текста по центру экрана или столбца таблицы;

**JUSTIFY** — выравнивание текста по ширине экрана или столбца таблицы.

Таким образом, текст, форматирование которого внутри абзаца задано тегом **<Р ALIGN=JUSTIFY>,** будет выровнен на экране компьютера по всей его ширине. Если при использовании тега <Р> не задано никаких атрибутов, по умолчанию используется левое выравнивание.

Можно не включать ни один из перечисленных выше аргументов в состав тега форматирования абзаца, поскольку язык разметки гипертекста позволяет использовать некоторые из них как самостоятельные **теги, предназначенные для позиционирования нескольких объектов документа в окне броузера**. Используются эти теги следующим образом:

**<LEFT>**

**Часть кода HTML**

**</LEFT>**

Все, что заключено между тегами <LEFT> и </LEFT> — текст, таблицы, графические объекты, — будет выровнено по левому краю экрана.

Если вы хотите, чтобы внутри текстового блока броузер отобразил перевод строки, например, для **отделения одного абзаца от другого**, в конце строки следует поместить непарный тег **<BR>**. Следующее слово, встречающееся за данным тегом, будет перенесено на новую строку. Примером использования этой команды может служить следующий отрывок кода HTML:

**<P>**

**Данный текст находится на одной строке экрана.<BR>**

**а этот текст отобразится уже с новой строки.**

**</Р>**

Использование нескольких тегов <BR> подряд, задает последовательность из нескольких пустых строк.

Иногда тег <BR> содержит атрибут **CLEAR**, который может принимать одно из трех значений: LEFT, RIGHT или ALL. Указанные параметры позволяют переносить текст со смещением новой строки соответственно к левой, правой или обеим границам экрана одновременно. Синтаксис записи этого тега с атрибутом CLEAR таков:

**<BR СLЕАР="значение">**.

Некоторые броузеры переносят строки самостоятельно, причем разрывают строку в том месте, где она выходит за границы экрана. Если программисту необходимо отобразить **текстовую строку фиксированной длины без переносов**, используется тег **<NOBR>:**

**<NORB>строка без переносов</NORB>**

При использовании данного тега в случае, если длина строки превышает ширину экрана, в нижней части окна броузера появляется горизонтальная **полоса прокрутки**.

Иногда для расстановки логических акцентов в тексте создателю web-страницы приходится использовать контекстное выделение каких-либо слов или выражений шрифтом: **жирным или курсивным**. В этих случаях также необходимо использовать соответствующие директивы HTML:

**<В>Текст. выделяемый жирным шрифтом</В>**

**<I>Текст, выделяемый курсивным шрифтом</I>**

Данные пары тегов можно вкладывать одна в другую. Например, выделить текст жирным курсивным шрифтом можно так:

**<В><I>Текст, выделенный жирным курсивным шрифтом</I></В>**

Однако создатели спецификации языка HTML предостерегают программистов от злоупотребления этими тегами, поскольку интерпретаторы далеко не всех современных броузеров умеют их правильно распознавать. Именно поэтому рекомендуется использование аналогичных директив, называемых **«командами логического выделения»**. Они выполняют те же функции, но синтаксис записи у них несколько иной:

**<ЕМ>Текст, выделяемый курсивным шрифтом</ЕМ>**

**<STRONG>Текст, выделяемый жирным шрифтом</STRONG>**

Очевидно, что для выделения текста жирным курсивным шрифтом можно использовать вложение данных тегов друг в друга:

**<STRONG> <ЕМ>Текст, выделенный жирным курсивным шрифтом**

**</ЕМ></STRONG>**

Для выделения текста подчеркиванием используется тег **<U>**. Синтаксис его записи также достаточно прост:

**<U>Подчеркнутый текст</U>**

Элемент подчеркивания может быть вложенным тегом, спецификация языка позволяет использовать его совместно с командами выделения текста жирным и курсивным шрифтом.

Помимо перечисленных тегов выделения текста существует еще несколько текстовых элементов, позволяющих различными методами акцентировать внимание зрителя на тех или иных фразах или словах. Среди подобных команд необходимо отметить следующие:

а) парный тег **<STRIKE>** позволяет отображать в окне броузера зачеркнутый текст.

б) парный тег **<SUB>** позволяет отобразить подстрочный текст. Синтаксис записи:

**текст 1<SUB>текст 2</SUB>**

Результат обработки такой строки кода будет иметь следующий вид: **текст1текст2**

в) парный тег **<SUP>** позволяет отобразить надстрочный текст. Синтаксис записи:

**текст1<SUP>текст 2</SUP>**

Результат обработки такой строки кода будет иметь следующий вид: **текст1текст2**

г) парный тег **<BIG>** — тег, позволяющий вывести часть текста крупным шрифтом.

**ПРИМЕЧАНИЕ** *При обработке кода HTML интерпретатором броузера размер шрифта высчитывается не в пикселах, а в условных пунктах нелинейного размера, величина такого «пункта» зависит от используемой вами реализации броузера. «Нормальный» (подставляемый по умолчанию) шрифт имеет высоту три пункта.*

Использование тега <BIG> увеличивает «нормальный» размер шрифта *на один пункт*. Синтаксис записи этой команды таков:

**<BIG>текст</BIG>**

Чтобы увеличить размер шрифта *более чем на один пункт*, можно использовать последовательность из нескольких тегов <BIG>:

**<BIG><BIG>тeкст</BIG></BIG>**

Аналогично уменьшения размера шрифта на один пункт можно добиться с использованием тега **<SMALL>**. Мнемоника записи этого тега выглядит следующим образом:

**<SMALL>текст</SMALL>**

Допустимо также использование нескольких вложенных элементов <SMALL>, подобно тому, как это было показано для тега <BIG>.

Следует помнить, что смысловое выделение текста лучше всего применять тогда, когда это действительно необходимо, поскольку текст, излишне пестрящий жирными, курсивными или подчеркнутыми словами, труден для восприятия, при его чтении сильно устают глаза и рассеивается внимание. Контекстное выделение можно использовать для отображения цитат или ключевых фраз, однако **большую часть текста лучше выводить стандартным шрифтом «нормального» размера без выделения**.

**10.** **Элемент <FONT>**

Директива **<FONT>** дает возможность изменять другие параметры шрифта.

Как известно, размер шрифта в стандарте HTML обозначается условными, пунктами, причем всего их насчитывается семь. Однако не все броузеры понимают термин «пункт» одинаково. Например, при отображении документа в Microsoft Internet Explorer размеры всех шрифтов различны, за исключением пунктов 2 и 3 — они отображаются шрифтом одной высоты, которая равна величине, принятой по умолчанию, и зависит от пользовательских настроек броузера. В Netscape Navigator абсолютно одинаково отображаются пункты 4 и 5, а так же 6 и 7, они больше по высоте, чем пункт 3, который является «нормальным» размером шрифта.

**ПРИМЕЧАНИЕ** *Следует помнить, что некоторые броузеры, например Microsoft InternetExplorer, позволяют настраивать параметры текста, отображаемого в составе документов HTML, путем изменения размера шрифта, подставляемого интерпретатором броузера по умолчанию. Это значит, что параметры элемента <FONT>, равно как и других тегов, управляющих размером шрифта, могут изменяться в зависимости от пользовательских настроек. По той же причине возможен эффект нарушения компоновки страницы, оптимизированной для просмотра в броузере со стандартными настройками.* *Избежать этого можно, заключив все элементы документа в невидимую таблицу с фиксированной шириной, заданной в пикселах.*

В общем виде синтаксис записи тега <FONT> выглядит следующим образом:

**<FONT FACE="значение1" SIZE="значение 2" COLOR="значение 3">текст</FONT>**

Значением **атрибута FACE** является **название шрифта**, который вы планируете использовать для отображения текста, заключенного в теги <FONT> </FONT>. Если, например, вместо параметра "значение1" в приведенном примере вы подставите "Arial", помеченный тегом <FONT>, текст будет отображаться именно этим шрифтом.

**ВНИМАНИЕ** *Используйте указания только на стандартные шрифты, имеющиеся в составе большинства броузеров: Arial и Times New Roman.*

**Атрибут SIZE** указывает **на размер шрифта**, которым следует выводить текст. Параметр этого атрибута, обозначенный в нашем примере как "значение2", может быть описан либо абсолютной, либо относительной величиной. Абсолютная величина подразумевает использование в качестве параметра целого числа в диапазоне от 1 до 6, указывающего на высоту шрифта в пунктах. Относительная же величина, обозначаемая целым числом со знаком плюс или минус (например, +2 или -1), — это количество пунктов, которые следует прибавить или отнять от размера шрифта, используемого броузером по умолчанию.

Так, запись **<FONT SIZE="+1">** говорит о том, что размер помеченного таким тегом текста будет на один пункт больше, чем обычный текст документа.

Вместо значения **атрибута COLOR** подставляется либо **цифровой код нужного цвета**, либо обозначающая этот цвет символьная метка.

Очевидно, что при использовании тега <FONT> любой из его атрибутов может быть опущен. Примером записи такой команды может служить следующая строка кода HTML:

**<FONT FACE="Arial" SIZE="+2" COLOR="#0000FF"> отрывок текста </FONT>**

**11. Тег <BODY>**

О теге <BODY>, обозначающем тело (основную часть) документа HTML, говорилось в начале. Необходимо добавить лишь то, что данная команда также может иметь атрибуты, использование которых в ряде случаев бывает весьма полезно, а иногда — просто необходимо.

В общем виде синтаксис записи тега <BODY> со всеми допустимыми атрибутами выглядит так: **<BODY BACKGROUND="URL" BGCOLOR="значение1" ТЕХТ="значение2" LINK="значение3" VLINK="значение4" ALINK="значение5"> тело документа HTML <BODY>**

**Атрибут BACKGROUND** позволяет дизайнеру поместить на web-страницу некий **фоновый рисунок**, записав в качестве параметра атрибута URL этого рисунка. URL можно задавать либо в виде полного адреса Интернета (например, "http://www.server.ru/imaes/имя\_файла.gif"), либо в виде сокращенного адреса с указанием пути к директории на текущем сервере, в которой хранится данное изображение (например, ".. images/имя\_файла.gif").

Допускается просто указывать имя графического файла, если он хранится в той же директории, что и использующий его файл HTML. Данное изображение может иметь любой размер, поскольку при интерпретации кода оно многократно повторяется, заполняя все доступное пространство в окне броузера.

**СОВЕТ** *Для того чтобы избежать неадекватности отображения того или иного цвета различными броузерами, например, когда web-дизайнер решил применить на странице заливку какого-либо редко используемого оттенка, рекомендуется следующий подход: создайте в любом подходящем редакторе графический файл нужного цвета размерами 1x1 пиксел, после чего укажите его в качестве фонового изображения, включив URL этого рисунка в параметр атрибута BACKGROUND тега <BODY>.*

**Атрибут TEXT** позволяет задать цвет текста для всего документа в целом. Но не забывайте, что параметр, назначенный данному атрибуту в теге <BODY>, можетбыть изменен в определенном участке текста путем использования команды<FONT> с атрибутом COLOR.

Для того чтобы назначить фоновый цвет всему документу, используется атрибут BGCOLOR. В этом случае web-страница будет целиком заполнена равномерной заливкой указанного цвета.

**ПРИМЕЧАНИЕ** Атрибуты BGCOLOR и BACKGROUND не исключают друг друга, однако у последнего имеется приоритет. Это означает, что в случае, когда заданы оба эти атрибута, сначала выполняется заливка web-страницы цветом, назначенным в атрибуте BGCOLOR, поверх которой размещается изображение, заданное атрибутом BACKGROUND.

**Атрибут LINK** дает web-мастеру возможность назначить цвет, которым отображается *не посещенная гиперссылка*, то есть ссылка, к которой посетитель данного web-сайта еще не обращался. По умолчанию ей присваивается значение **"blue" (#0000FF)**.

**Атрибут VLINK** указывает на цвет *посещенной ссылки*, значение которой по умолчанию — **"purple" (#800080)**.

**Атрибут** **ALINK** определяет *цвет активной гиперссылки*, то есть цвет, который гиперссылка принимает с момента нажатия на нее курсором мыши до момента загрузки вызываемого ею ресурса. По умолчанию данный атрибут также имеет значение **"purple"**.

Очевидно, что значения всех атрибутов тега <BODY>, кроме атрибута BACKGROUND, представляют собой обозначения цветов символьными метками или шестнадцатеричным цифровым кодом.

**ВНИМАНИЕ** Установив одно из значений цветовых параметров вашей страницы, жестко задавайте и остальные значения. Некоторые броузеры позволяют пользователю произвольно менять цвет фона документа или выводимого на экран текста, если они не указаны явно. В этом случае заданный вами цвет текста может совпасть с цветом фона, установленным в броузере пользователя по умолчанию, в результате чего текст станет нечитаемым. Изменение одного из цветовых параметров без изменения остальных недопустимо. Исключение можно сделать лишь в том случае, когда в качестве фонового цвета web-страницы используется белый.

Вот пример использования тега <BODY> со всеми допустимыми атрибутами:

**<BODY BACKGROUND="http://www.myserver.com/images/back.jpg" BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#008000" VLINK="#800080" ALINK="#FF0000"> Тело документа HTML </BODY>**

**12. Заголовки**

Спецификация HTML позволяет программисту использовать шесть различных стилей, или логических уровней заголовков. Для того чтобы отобразить заголовок на web-странице, необходимо использовать тег **<Нn>**, где n — целое число от 1 до 6, обозначающее номер уровня заголовка, где самым **верхним уровнем** принято считать **1**.

Синтаксис записи этой команды в общем виде таков:

**<Нn АLIGN="параметр">Текст заголовка</Нn>**

**Атрибут ALIGN** дает возможность web-мастеру определить расположение заголовка в окне броузера. Параметр этого атрибута может принимать одно из трех возможных значений: RIGHT, LEFT, CENTER. Вот пример использования данного тега:

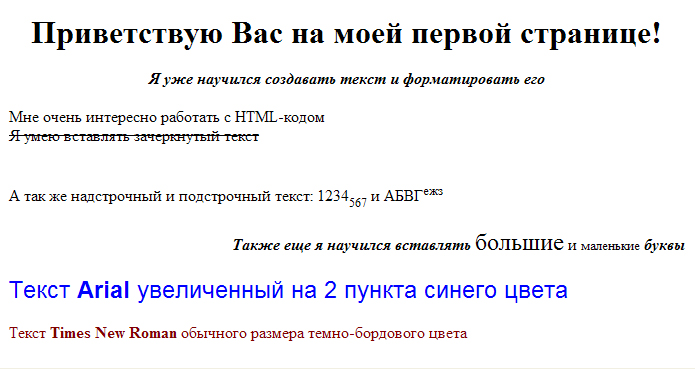
**<Н1 ALIGN="CENTER">ДOБPO ПОЖАЛОВАТЬ НА МОЮ СТРАНИЧКУ! </Н1>**

Не рекомендуется включать в документы заголовки ниже уровня <Н4>, поскольку они отображаются значительно более мелким шрифтом, нежели основной текст web-страницы, что затрудняет восприятие информации. Во всех остальных случаях использование данной команды — дело вкуса и эстетических предпочтений автора проекта.

**Задания для практического занятия:**

1. Создайте новую директорию (папку) на жестком диске компьютера и назовите ее, например, «Site\_vasha\_familiya». Внутри этой папки создайте каталог с названием «images», туда вы будете впоследствии помещать иллюстрирующие ваш проект изображения.

2. В созданном вами файле с помощью программы Блокнот запишите следующий текст:

1

3. Вставьте в код своей страницы текст комментария, например предупреждение об авторских правах.

4. Создайте заголовок первого уровня (тег **<H1>**) и «отцентрируйте» его.

5. Создайте маркированный (**<ul> <li> </li> </ul>**) и нумерованный (**<ol> <li> </li> </ol>**) списки.

Виды операционных систем:

* + 1. Unix
    2. Linux
    3. FreeBSD
    4. **OS/2**
    5. MacOS
    6. VMSP5
    7. SunOS
    8. VMESA
    9. Windows

Активный контент:

* + Java
  + ActiveX
  + JavaScript
  + VBScript
  + Action Script
  + Coockies

6. Создайте гипертекстовую ссылку (тег **<a href="/"> </a>**) с последнего пункта нумерованного списка.

7. Поместите изображение на страницу (тег **<img src="/" />**).

8. Сделайте размещенное изображение ссылкой (тег **<a href="/"> </a>**).

9. Еще раз разместите это же изображение на странице, и также сделайте его гипертекстовой ссылкой. Добейтесь, чтобы рамка вокруг изображения не была видна. Отцентрируйте изображение и текст под ним.

10. Сделайте задний фон страницы светло-желтым \_\_ (#FFFFD9),   
цвет непосещенных ссылок – красным \_\_ (#FF0000),   
посещенных – синим \_\_ (#0000FF).

11. В нумерованном списке текст пункта 4 сделайте **жирным** (теги **<strong> </strong>**и **<b> </b>**), а пункта 5 - *курсивным* (наклонным) (теги **<em> </em>**и **<i> </i>**).

12. Оформите в абзац (тег **<p> </p>**) текст, предстваленный ниже. Словосочетание «гипертекстовая ссылка» оформите зеленым цветом (#008000)\_\_

Что такоегипертекстовая ссылка(или, как чаще говорят, просто ссылка)?  
С помощью ссылки устанавливается связь с другим (или с тем же) документом. Ссылкой может быть фрагмент текста или изображение. По умолчанию текстовая ссылка подчеркнута. При подведении курсора мыши к гипертекстовой ссылке он превращается в раскрытую ладонь. При нажатии на ссылку браузер обращается к серверу, запрашивая документ, который указан в html-теге ссылки.

13. Создайте абсолютную ссылку на сайт www.mail.ru

*Для поиска информации воспользуйтесь поисковой машиной*[*www.mail.ru*](http://www.mail.ru/)

14. Сохраните страницу в личной папке в файл firstpage.html

**Контрольные вопросы**

1. Дайте определение понятия тег.

2. Структура HTML-документа.

3. Какие виды тегов вы знаете?

**Практическая работа № 2**

**«Использование таблиц при создании web-страниц»**

**Цель работы:** научиться применять табличную разметку HTML-документа.

**Образовательные результаты, заявленные во ФГОС третьего поколения:**

Студент должен

уметь:

- составлять HTML-код страницы;

- проектировать web-приложения;

знать:

- синтаксис HTML.

**Краткие теоретические и учебно-методические материалы по теме практической работы**

**HTML-файл** содержит один тип управляющих конструкций - *теги (tags).*

*тег —* единица разметки;

*элемент —* составная часть документа.

**Назначение тегов:**

- разделяютисходный неформатированный текст документа на элементы;

- создаютновые элементы, которым ничего не соответствовало в тексте (например, графические вставки или Java-апплеты).

**Виды тегов:**

- *парные,* охватывающие какой-то фрагмент текста и/или другие теги,

***<парный-тег>текст или другие теги</парный-тег>***

Парные теги должны вкладываться друг в друга без пересе­чений, т. е. если в области действия тега А открылся тег В, он должен закрыться до того, как закроется тег А.

***<A> <B> </B> </A>***

- *непарные,* стоящие в одиночестве*:*

***<непарный-тег>***

**Атрибуты**

Многие теги, как парные, так и непарные, имеют *атрибуты,* изменяющие и уточняющие действие тега:

***<тег атрибут1="значение" атрибут2="значение" ...>***

Регистр букв в идентификаторах тегов и атрибутов (но не в значениях атрибутов) не учитывается. Пары **атрибут="значение"** распознаются как таковые только внутри угловых скобок тега и отделяются друг от друга пробелами.

**Подстановки**

Чтобы ввести в документ символы, от­сутствующие на клавиатуре или же имеющие в синтаксисе HTML специальное значение, употребляются *подстановки* двух видов:

1. *мнемонические -*  имеют вид

**& *мнемонический код;***

например:

&egrave; для ё  
&lt; для <

&nbsp; для символа неразрываемого пробела

2. *числовые –* имеют вид

**& # *десятичный числовой код;***

например:

&#1б0; для символа неразрываемого пробела

**Ссылки и привязки**

**тег А**

Любая ссылка в HTML имеет два обязатель­ных элемента:

источник - то изображение или фрагмент, который заключен между <А> и </А> и щелчок по которо­му активизирует ссылку;

пункт назначения - URL-адрес документа, на который ведет ссылка.

Адрес назначения может указывать не только на весь документ в целом, но и на какое-то место (точнее, опять-таки, какой-то *элемент)* внутри документа, в том числе и внутри самого документа со ссылкой. Для этого пункт назначения должен быть помечен с помощью атрибута name тега А создателем того документа, на который делается ссылка. В свою очередь, в теге А в документе-источнике эта метка приписывается к адресу назначения через символ #.

**Формы**

**тег FORM**

*Формы (forms)*, или бланки, предназначены для «обратной связи», т.е. отсылки информации от пользователя обратно на сервер. Набор органов управления HTML-бланков соответ­ствует возможностям современных графических операци­онных систем и включает в себя поля для ввода текста (однострочные и многострочные), флажки, переключатели, выпадающие списки, списки с прокруткой и кнопки.

Тег FORM, объединяет группу связанных по смы­слу элементов и указывает адрес той программы на сервере, которой будут посланы введенные пользователем данные из всех элементов формы. HTML-страница может содержать любое количество независимых друг от друга форм, в каждой из которых должна присутствовать «пуско­вая кнопка», отправляющая данные на сервер. Кнопке этой не обязательно быть стандартной интерфейсной кнопкой, в этой же роли может использоваться изображение, а для простых форм, состоящих из одного поля ввода или выпа­дающего списка, посылка данных может активизироваться нажатием Enter в поле ввода или операцией выбора элемента в списке.

**Изображения и объект**

**Тег IMG**

Тег IMG, предназначенный длявставки изображений, относится к тегам, создающим но­вые элементы документа, отсутствовавшие в исходном тексте. Тег этот ссылается на хранящееся в отдельном файле изображение в формате GIF или JPEG; этот графический файл может располагаться там же, где и HTML-файл страницы (в таком случае в *атрибуте* src достаточно указать имя файла), а может лежать и в дру­гом каталоге и даже на другом сервере (в этом случае нужно указывать полный URL-адрес). Большинство атри­бутов этого тега управляют форматированием изображения, устанавливая его размеры, поля, выравнивание и проч.

**Таблицы**

**Тег TABLE**

Тег TABLE чаще используется для визуального форматирования страницы, чем для представления табличного по своей природе материала.

Используются *атрибуты* height, width, cellspacing, cellpadding, border для форматирования внешнего вида таблицы.

**Задания для практического занятия:**

# **Размещение таблиц на html-странице. Форматирование текста в таблице**

1. Создайте новую HTML-страницу.
2. Создайте новую Web-страницу. Разместите на странице таблицу.  
   Ширина таблицы = 90%, высота = 600 пикселей.  
   Высота рядов = 200 пикселей.  
   Ширина колонок = 30%  
   Выполните форматирование текста в таблице, как показано в **примере**

### *В таблице могут использоваться следующие способы выравнивания:*

|  |  |  |
| --- | --- | --- |
| **По горизонтали:** - слева; **По вертикали:** - вверху. | **По горизонтали:** - слева; **По вертикали:** - по центру. (этот способ используется по умолчанию) | **По горизонтали:** - слева; **По вертикали:** - внизу. |
| **По горизонтали:** - по центру; **По вертикали:** - вверху. | **По горизонтали:** - по центру; **По вертикали:** - по центру. | **По горизонтали:** - по центру; **По вертикали:** - внизу. |
| **По горизонтали:** - справа; **По вертикали:** - вверху. | **По горизонтали:** - справа; **По вертикали:** - по центру. | **По горизонтали:** - справа; **По вертикали:** - внизу. |

1. Сохраните страницу в личной папке в файл table1.html

# **2.2 Размещение таблиц на html-странице. Форматирование ячеек таблицы**

1. Создайте новую HTML-страницу.
2. Создайте новую Web-страницу. Разместите на странице таблицу.  
   Выровняйте таблицу по центру страницы.  
   Цвет заднего фона таблицы = "#E2E2E2".  
   Ширина таблицы = 500 пикселей.  
   Расстояние между границей ячейки и ее содержимым = 15 пикселей.  
   Толщина границы = 5 пикселей, цвет границы = "#008000  
   Цвет второго ряда таблицы = #FFFFCA"  
   Цвет ячейки 3 = "#FFCACA", цвет ячейки 5 = "#ECFFEC" (см. **пример** ).

### *Форматирование заднего фона таблиц, рядов и ячеек:*

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

1. Сохраните страницу в личной папке в файл table2.html

# **2.3 Размещение таблиц на html-странице. Объединение ячеек таблицы**

1. Создайте новую HTML-страницу.
2. Создайте новую Web-страницу.  
   Разместите на странице таблицу из 5 рядов и 5 колонок.  
   Выполните объединение ячеек таблицы, как показано в **примере**

### *Объединение ячеек таблицы:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | |  |
|  | | |
|  |  |  |
|  |  |  | |
|  |

1. Сохраните страницу в личной папке в файл table3.html

**Контрольные вопросы**

1.С помощью каких атрибутов тега «table» объединяются ячейки?

2.Как задается расстояние между границей ячейки и ее содержимым?

3.Какие атрибуты тега «table» вы применяли в данной практической работе? Что они обозначают?

**Практическое занятие 3  
  
 «Дополнительные элементы языка HTML для форматирования web-страниц»**

**Цель работы:** Научиться применять тег «Горизонтальные линии (тег <HR>)», "Escape-поcледовательности" и «Физические и логические стили».

**Образовательные результаты, заявленные во ФГОС третьего поколения:**

Студент должен

уметь:

- составлять HTML-код страницы;

- проектировать web-приложения;

знать:

- синтаксис HTML.

**Краткие теоретические и учебно-методические материалы по теме практической работы**

**HTML-файл** содержит один тип управляющих конструкций - *теги (tags).*

*тег —* единица разметки;

*элемент —* составная часть документа.

**Назначение тегов:**

- разделяютисходный неформатированный текст документа на элементы;

- создаютновые элементы, которым ничего не соответствовало в тексте (например, графические вставки или Java-апплеты).

**Виды тегов:**

- *парные,* охватывающие какой-то фрагмент текста и/или другие теги,

***<парный-тег>текст или другие теги</парный-тег>***

Парные теги должны вкладываться друг в друга без пересе­чений, т. е. если в области действия тега А открылся тег В, он должен закрыться до того, как закроется тег А.

***<A> <B> </B> </A>***

- *непарные,* стоящие в одиночестве*:*

***<непарный-тег>***

**Атрибуты**

Многие теги, как парные, так и непарные, имеют *атрибуты,* изменяющие и уточняющие действие тега:

***<тег атрибут1="значение" атрибут2="значение" ...>***

Регистр букв в идентификаторах тегов и атрибутов (но не в значениях атрибутов) не учитывается. Пары **атрибут="значение"** распознаются как таковые только внутри угловых скобок тега и отделяются друг от друга пробелами.

**Подстановки**

Чтобы ввести в документ символы, от­сутствующие на клавиатуре или же имеющие в синтаксисе HTML специальное значение, употребляются *подстановки* двух видов:

1. *мнемонические -*  имеют вид

**& *мнемонический код;***

например:

&egrave; для ё  
&lt; для <

&nbsp; для символа неразрываемого пробела

2. *числовые –* имеют вид

**& # *десятичный числовой код;***

например:

&#1б0; для символа неразрываемого пробела

**Ссылки и привязки**

**тег А**

Любая ссылка в HTML имеет два обязатель­ных элемента:

источник - то изображение или фрагмент, который заключен между <А> и </А> и щелчок по которо­му активизирует ссылку;

пункт назначения - URL-адрес документа, на который ведет ссылка.

Адрес назначения может указывать не только на весь документ в целом, но и на какое-то место (точнее, опять-таки, какой-то *элемент)* внутри документа, в том числе и внутри самого документа со ссылкой. Для этого пункт назначения должен быть помечен с помощью атрибута name тега А создателем того документа, на который делается ссылка. В свою очередь, в теге А в документе-источнике эта метка приписывается к адресу назначения через символ #.

**Формы**

**тег FORM**

*Формы (forms)*, или бланки, предназначены для «обратной связи», т.е. отсылки информации от пользователя обратно на сервер. Набор органов управления HTML-бланков соответ­ствует возможностям современных графических операци­онных систем и включает в себя поля для ввода текста (однострочные и многострочные), флажки, переключатели, выпадающие списки, списки с прокруткой и кнопки.

Тег FORM, объединяет группу связанных по смы­слу элементов и указывает адрес той программы на сервере, которой будут посланы введенные пользователем данные из всех элементов формы. HTML-страница может содержать любое количество независимых друг от друга форм, в каждой из которых должна присутствовать «пуско­вая кнопка», отправляющая данные на сервер. Кнопке этой не обязательно быть стандартной интерфейсной кнопкой, в этой же роли может использоваться изображение, а для простых форм, состоящих из одного поля ввода или выпа­дающего списка, посылка данных может активизироваться нажатием Enter в поле ввода или операцией выбора элемента в списке.

**Изображения и объект**

**Тег IMG**

Тег IMG, предназначенный длявставки изображений, относится к тегам, создающим но­вые элементы документа, отсутствовавшие в исходном тексте. Тег этот ссылается на хранящееся в отдельном файле изображение в формате GIF или JPEG; этот графический файл может располагаться там же, где и HTML-файл страницы (в таком случае в *атрибуте* src достаточно указать имя файла), а может лежать и в дру­гом каталоге и даже на другом сервере (в этом случае нужно указывать полный URL-адрес). Большинство атри­бутов этого тега управляют форматированием изображения, устанавливая его размеры, поля, выравнивание и проч.

**Таблицы**

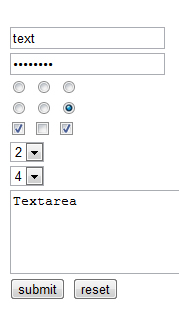
**Тег TABLE**

Тег TABLE чаще используется для визуального форматирования страницы, чем для представления табличного по своей природе материала.

Используются *атрибуты* height, width, cellspacing, cellpadding, border для форматирования внешнего вида таблицы.

Тег **<FORM>** используется для создания форма на web-странице. Форма служат для отправки данных на сервер и обеспечивают диалог пользователя с приложением.   
Для отправки формы на сервер обычно используется кнопка, созданная при помощи тега **<BUTTON>** или **<INPUT>**, иногда этого же можно добиться, если нажать клавишу **<Enter>**, находясь в одном из полей данных формы.  Когда форма отправляется на сервер, управление данными передается CGI-программе или документу, определенному параметром **action** тега **<FORM>**. Предварительно браузер подготавливает информацию в виде пары «имя=значение», где имя определяется параметром **name** тега **<INPUT>**, а значение введено пользователем или установлено в поле формы по умолчанию. Если для отправки данных используется метод **GET**, то адресная строка может принимать вид:  **http://webcreature.ru/index.php?user=Alex&button=activate**  Все параметры в адресной строке в этом случае, перечисляются послезнака "**?**", а символ "**&**" служит для них связкой.

Пример:

<form name="test\_form\_name" id="test\_form\_id" method="get" action="?">  
<input type="hidden" name="hidden\_name" id="hidden\_id" value="hidden" /><br/>  
<input type="text" name="text\_name" id="text\_id" value="text" /><br/>  
<input type="password" name="password\_name" id="password\_id" value="password" /><br/>  
  
<input type="radio" name="radio\_group\_1" id="radio\_id\_1" value="radio\_1" />  
<input type="radio" name="radio\_group\_1" id="radio\_id\_2" value="radio\_2" />  
<input type="radio" name="radio\_group\_1" id="radio\_id\_3" value="radio\_3" /><br/>  
  
<input type="radio" name="radio\_group\_2" id="radio\_id\_a" value="radio\_a" />  
<input type="radio" name="radio\_group\_2" id="radio\_id\_b" value="radio\_b" />  
<input type="radio" name="radio\_group\_2" id="radio\_id\_c" value="radio\_c" checked="checked" /><br/>  
  
<input type="checkbox" name="check\_a" id="check\_id\_a" value="check\_1" checked="checked" />  
<input type="checkbox" name="check\_b" id="check\_id\_b" value="check\_2" />  
<input type="checkbox" name="check\_c" id="check\_id\_c" value="check\_3" checked="checked" /><br/>  
  
<select name="select\_name\_from" id="select\_id\_from">  
<option value="option\_1">1</value>  
<option value="option\_2" selected="selected">2</value>  
<option value="option\_3">3</value>  
</select><br/>  
<select name="select\_name\_to" id="select\_id\_to">  
<option value="option\_4">4</value>  
<option value="option\_5">5</value>  
<option value="option\_6">6</value>  
</select><br/>  
<textarea name="textarea\_name" id="textarea\_id" cols="20" rows="5">Textarea</textarea><br/>  
  
<input type="submit" name="submit\_name" id="submit\_id" value="submit" />  
<input type="reset" name="reset\_name" id="reset\_id" value="reset" />  
</form>

**Задания для практического занятия:**

1. Создайте новую HTML-страницу, состоящую из 7-ми частей.

2. Отделите каждую часть горизонтальной линией (тег <HR>) с разными параметрами, как показано в примере

### *Коллекция горизонтальных линий*

Часть первая

Часть вторая

Часть третья

Часть четвёртая

Часть пятая

Часть шестая

Часть седьмая

3. Сохраните страницу в личной папке в файл rows.html

4. Создайте новую HTML-страницу.

5. Разместите на странице таблицу.

6. Отформатируйте страницу с использованием таблиц и Escape–последовательностей как указано в примере

### *Escape-поcледовательности*

Для создания Web-страниц, содержащих специальные символы, можно использовать Escape-последовательности.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Символ | Числовая подстановка | Мнемоническая подстановка |
| Неразрывный пробел |  | &#160; | &nbsp; |
| Перевернутый восклицательный знак | ¡ | &#161; | &ixcl; |
| Цент | ¢ | &#162; | &cent; |
| Фунт | £ | &#163; | &pound; |
| Валюта | ¤ | &#164; | &curren; |
| Йена | ¥ | &#165; | &yen; |
| Прерванный штрих | ¦ | &#166; | &brvbar; |
| Параграф/раздел | § | &#167; | &sect; |
| Toчки умлаута | ¨ | &#168; | &uml; |
| Копирайт | © | &#169; | &copy; |
| Порядковое числительное женского рода | ª | &#170; | &ordf; |
| Французская кавычка слева | « | &#171; | &laquo; |
| Не/Нет | ¬ | &#172; | &not; |
| Дефис |  | &#173; | &shy; |
| Зарегестрированный торговый знак | ® | &#174; | &reg; |
| Надчеркивание | ¯ | &#175; | &macr; |
| Градус | ° | &#176; | &deg; |
| Плюс-минус | ± | &#177; | &plusmn; |
| Вторая степень | ² | &#178; | &sup2; |
| Третья степень | ³ | &#179; | &sup3; |
| Маленькое а с ударением | ´ | &#180; | &acuate; |
| Мю | µ | &#181; | &micro; |
| Параграф | ¶ | &#182; | &para; |
| Средняя точка | · | &#183; | &middot; |
| Цедиль | ¸ | &#181; | &cedil; |
| Первая степень | ¹ | &#185; | &sup1; |
| Порядковое числительное мужского рода | º | &#186; | &ordm; |
| Французская кавычка справа | » | &#187; | &raquo; |
| Дробь 1/4 | ¼ | &#188; | &frac14; |
| Дробь 1/2 | ½ | &#189; | &frac12; |
| Дробь 3/4 | ¾ | &#190; | &frac34; |
| Перевернутый вопросительный знак | ¿ | &#191; | &iquest; |

7. Создайте новую HTML-страницу.

8. Отформатируйте с помощью тегов логического и физического форматирования, как показано в **примере**

### *Физические и логические стили*

#### § 1 - использование тегов <sup>...</sup>, <sub>...</sub>(верхний и нижний индекс)

В 2002 году компания Microsoft® выпустила операционную систему Windows XPTM

*f3=x4\*(a3+b6)*

#### § 2 - использование кавычек

© Лев Толстой написал книгу «Война и Мир».

#### § 3 - испльзование моноширинного шрифта (как на пишущей машинке)

В офисе секретарь печатал на пишущей машинке

#### § 4 - использование тега <code>...</code> (для записи фрагментов исходных текстов)

3²=9;  
3³=27;  
S=h/v;  
72\*100=7200 м²;  
72\*100\*50=360000 м³

#### § 5 - использование тега <blockquote>...</blockquote> (текст с отступом слева)

Стоимость процессора Intell© Pentium VI - 106 €

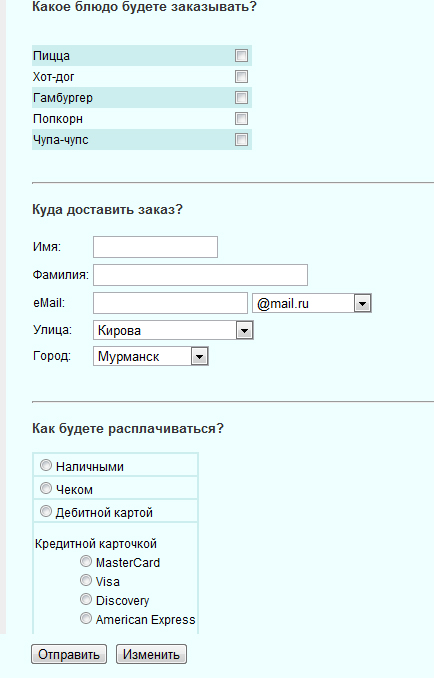
#### § 6 - использование тега <pre>...</pre> (предварительно форматированный) текст

**Ведомость на выдачу зарплаты**  
+-------+-----------------------+---------------+  
! **№ п.п.**!  **Фамилия И.О.**  !  **Зарплата**  !  
+-------+-----------------------+---------------+  
! 1. ! Иванов И.И. ! 2500 руб. !  
+-------+-----------------------+---------------+  
! 2. ! Петров П.П. ! 3150 руб. !  
+-------+-----------------------+---------------+  
! 3. ! Сидоров С.С. ! 1200 руб. !  
+-------+-----------------------+---------------+  
! ***Итого:*** ! **6850 руб.** !  
+-------+-----------------------+---------------+

#### § 7 - использование тега <marquee>...</marquee> (бегущая строка)

9. Создайте новую HTML-страницу.

* + 1. Создайте документ с несколькими формами (см. **пример**).



* 1. Сохраните страницу в личной папке в файл forma.html

**Контрольные вопросы**

1. Что такое Escape-последовательности?

2. Какие параметры тега горизонтальная линия вы знаете?

3. Какие вы знаете физические и логические стили?

4. .Как задаются списки переключателей?

5. С помощью каких тегов создается форма с использованием упорядоченных списков для нумерации полей ввода?

6.Для чего необходим тег **<FORM>**?

**Практическая работа № 4**

**«Форматирование web-страниц с использованием таблиц стилей.   
Создание каскадных листов стилей (css)»**

**Цель:** Изучить основы технологии CSS на примере разработки собственного сайта.

**Образовательные результаты, заявленные во ФГОС третьего поколения:**

Студент должен

уметь:

- составлять HTML-код страницы;

- проектировать web-приложения;

знать:

- синтаксис HTML.

**Краткие теоретические и учебно-методические материалы по теме практической работы**

**Стиль** — это набор правил оформления элемента web-стра­ницы.

**Селектор** — элемент стиля, в селекторе указаны параметры форматирования (цвет элемента, фоновый цвет, гарнитура шриф­та, размер символов, отступы, стили границ и др).

**Типы селекторов:** селекторы тегов, классы и идентификаторы.

Синтаксис стиля:

селектор {параметр].: значение;

параметр2: значение]., значение2;

}

При написании стиля можно использовать любой регистр клавиатуры, пробелы необязательны, значение можно помещать в двойные кавычки, значок // обозначает начало комментария.

**Варианты размещения CSS**

**1. В отдельном файле,** если одинаковым стилем оформлены разные страницы. Стили записываются в текстовый файл с рас­ширением CSS (например, mystyle.css):

**Пример:**

р { color: gold;

background-color: #990000;

}

В коде страницы, на которой используется стиль из этого файла, делают ссылку: **Пример:**

<head>

<link rel="stylesheet" type="text/css"

href="mystyle.css">

</head>

Стиль применяется, когда используют тег, оформленный данным стилем:

<body>

<р>текст</р>

</body>

2. **В** разделе ***head* в теге *style,*** если стиль на странице исполь­  
зуется несколько раз или в разных тегах.

**Пример:**

<head>

<style type="text/css"> р { color: gold; background-color: #990000;

}

</style> </head>

**3. В любом теге в атрибуте *style,*** если стиль используется редко.  
**Пример:**

<р style="color: gold; background-color: #990000;">

текст </р>

***Селекторы тегов.*** Любой тег HTML может быть селектором. Синтаксис:

*Тег { Параметр: Значение; }*

**Пример:**

а) <style>

body { text-align: justify; color: black; font-family:

Arial; }

</style>

б) <headxstyle>

HI { font-family: Arial, Helvetica, Verdana,

sans-serif; font-size: 150%; font-weight: light }

</style></head>

<body>

<Н1>Заголовок</Н1> Обычный текст

</body>

***Классы****.* Классы используют, когда нужно применить стиль к разным тегам web-страницы: ячейкам таблицы, ссылкам, па­раграфам и др.

Синтаксис класса:

Тег.Имя класса { Параметр: Значение; }

В тег добавляется атрибут *class= "Имя класса".* **Пример:**

<headxstyle>

P.cite ( color: navy; font-size: 80% margin: 20px }

// параграф с классом cite

</stylex/head>

<body>

<p> текст </р>

<p class=cite>TeKCT</p>

</body>

Если для класса не указывать конкретный тег, тогда класс можно применять к любому тегу. Синтаксис класса:

.Имя класса { Параметр: Значение; }

**Пример:**

<headxstyle>

.mycolor { color: navy; color-background: yellow; }

</style></head>

<h2 class=mycolor>TeKCT</h2>

<tablextr>

<td class=mycolor> текст </td>

<td>TeKCT</td>

</tr></table>

Тег ***span***используют для создания выделенного текста, бук­виц, цитат и др. **Пример:**

<headxstyle>

.capital { font-size: 150%; color: red; }

</styleX/head>

<body>

<span class=capital>B</span>yKBnua

</body>

*Псевдоклассы.* Псевдокласс используется только для оформ­ления ссылок. Виды псевдоклассов: непосещенная ссылка *(link),* состояние ссылки под курсором мыши *(hover),* состояние ссылки в момент щелчка *(active),* посещенная ссылка *(visited).* Синтак­сис псевдокласса: *А:псевдокласс { Параметр: Значение; }*

**Пример:**

<headxstyle type="text/css">

a:link { color: #003366; }

a:visited { color: #660066; }

a:hover { color: #800000; }

a:active ( color: #FF0O0O; }

</stylex/head>

<body>

| <a href=#>CcbinKa K/a> I <a href=#>Ссылка 2</a> I

<a href=#>CcbtnKa 3</a> I

</body>

**Пример:**

<head> <style>

a.linkl { font-size: 12px; color: green }

a.linkl:hover { color: red }

a.link2 {font-size: 14px; color: blue }

a.Iink2:hover { color: red }

</style> </head>

<body link=#0000ff>

I <a href=linkl.html>CcbinKa 1</а> | <a href=link2.html

class=linkl>CcbmKa 2</a> | <a href=link3.html

class=link2>CcbLnKa 3</a> |

</body>

**Пример** использования наиболее востребованного псевдо­класса *(hover):*

<headxstyle type="text/css">

a {color: #003366;} a:hover {color: #800000;}

</style></head>

**Пример** использования различных классов для разных **видов** ссылок:

<head> <style>

a.linkl {font-size: 12px; color: green;}

a.linkl:hover { color: red;}

a.link2 {font-size: 14px; color: blue;}

a.Iink2:hover {color: red;}

</style> </head>

<body>

I <a href=linkl.html>CcbuiKa 1</а> | <a href=link2.html

class=linkl>CcbmKa 2</a> | <a href=link3.html

class=link2>CcbuiKa 3</a> |

</body>

***Шрифт.***

**1. Семейства шрифтов font-family**

*Таблица 4.1.* **Примеры шрифтов, входящих в различные семейства шрифтов**

|  |  |  |
| --- | --- | --- |
| Семейство шрифтов | Примеры шрифтов | Описание семейства |
| serif | Times New Roman, Book Antiqua | Шрифт с засечками |
| sans-serif | Helvetica, Arial, Verdana | Шрифт без засечек |
| cursive | Zapf-Chancery | Курсивный шрифт |
| fantasy | Western, Comic | Декоративный шрифт |
| monospace | Courier New | Моноширинный шрифт |

Варианты шрифтов разделяются запятыми, название шриф­та, состоящее из нескольких слов, заключается **в** кавычки:

р {font-family: 'Western, Times New Roman', serif;}

**2. Толщина шрифта font-weight:** *lighter, bold, bolder.***Пример:**

p.bold {font-weight: bold;} (относительно предыдущего текста).

**3. Размер шрифта font-size:** относительный (проценты — %), или абсолютный (пикселы — рх, пункты — pt, сантиметры — cm и миллиметры — mm).  
  
**Пример:**

hi {font-size: 200%;} (относительно основного текста)

h2 {font-size: 50px;}

h3 {font-size: 12pt; }

**4. Оформление текста text-decoration. Значения: подчеркива­ние *(underline),* надчеркивание *(overline),* зачеркивание *(line-through),* нет подчеркивания (попе).**

**Примеры:**

h4 {text-decoration: underline;}

р {text-decoration: overline;}

.wrong {text-decoration: line-through;}

a {text-decoration: none;}

***Цвет.***

**1. Цвет элемента color — цвет элемента в модели RGB или по стандартным названиям цветов.**

**Примеры:**

hi {color: yellow;}

hi {color: yellow;}

.active {color: #FFFF00;}

2. **Цвет фона background-color — цвет фона элемента (фоно­вый цвет).**

**Пример:**

hl.grayback {background-color: #CCCCCC;}

***Отступы.***

**1. Выравнивание text-align — выравнивание абзаца относи­тельно страницы или ячейки таблицы. Значения: по левому краю *(left),* по центру *(center),* по правому краю *(right),* по шири­не *(justify).***

**Пример:**

р {text-align: justify;}

2. **Отступ первой строки text-indent — отступ первой строки абзаца (красная строка). Значение задается в пикселах (рх) или в пунктах (pt).**

**Пример:**

р {text-indent: 50pt;}

3. **Межстрочные интервалы Hne-heighn** — междустрочный ин­тервал абзаца. Значение задается в пикселах (рх), пунктах (pt) или в процентах от размера шрифта (%).

Пример:

р {line-height: 50%;} межстрочные интервалы.

**4. Отступы вокруг элемента margin** — отступы вокруг элемен­та. Значение задается в пикселах (рх) или в пунтках (pt). Различ­ные типы этого параметра: одинаковый отступ со всех сторон (margin), отступ слева (margin-left), отсуп справа (margin-right), отступ сверху (margin-top), отступ снизу (margin-bottom).

Примеры:

р {margin: ЗОрх;}

img {margin-left: 20px;}

***Рамка.***

**1. Толщина рамки border-width** — толщина границы. Различ­ные типы этого параметра: толщина левой границы *{border-left-width),* толщина верхней границы *{border-top-width),* толщина правой границы *{border-right-width),* толщина нижней границы *{border-bottom-width).* Значение задается в пикселах (рх).

**Пример:**

.lineTop {border-top-width: Зрх;}

*2.* **Цвет рамки border-color** — цвет рамки (границы).  
**Пример:**

.lineRed {border-color: #FF0000;}

3. **Стиль рамки border-style** — стиль границы. Значения: гра­ницы нет *{попе),* сплошная линия *{solid),* пунктирная линия*{dotted),* штриховая линия *{dashed),* двойная линия *{double),* объ­емная канавка *{groove),* объемный гребень *{ridge),* объемнаякнопка внутрь *{inset),* объемная кнопка наружу *{outset).*

**Пример:**

.lineSimple {border-style: outset;}

**4. Обобщающий стиль рамки border** — обобщающий стиль границы, объединяющий три параметра: толщину, стиль и цвет границы. Порядок записи параметров неважен:

*border:* ширина стиль цвет; *border-tor:* ширина стиль цвет; *border-right:* ширина стиль цвет; *border-bottom:* ширина стиль цвет; *border-left:* ширина стиль цвет;

**Пример:**

td.blueBorder {border: 4px dotted #003399}

Обобщающий стиль можно задавать для определенной сто­роны границы: слева (border-left), сверху (border-top), справа (border-right), снизу (border-bottom).

**Пример:**

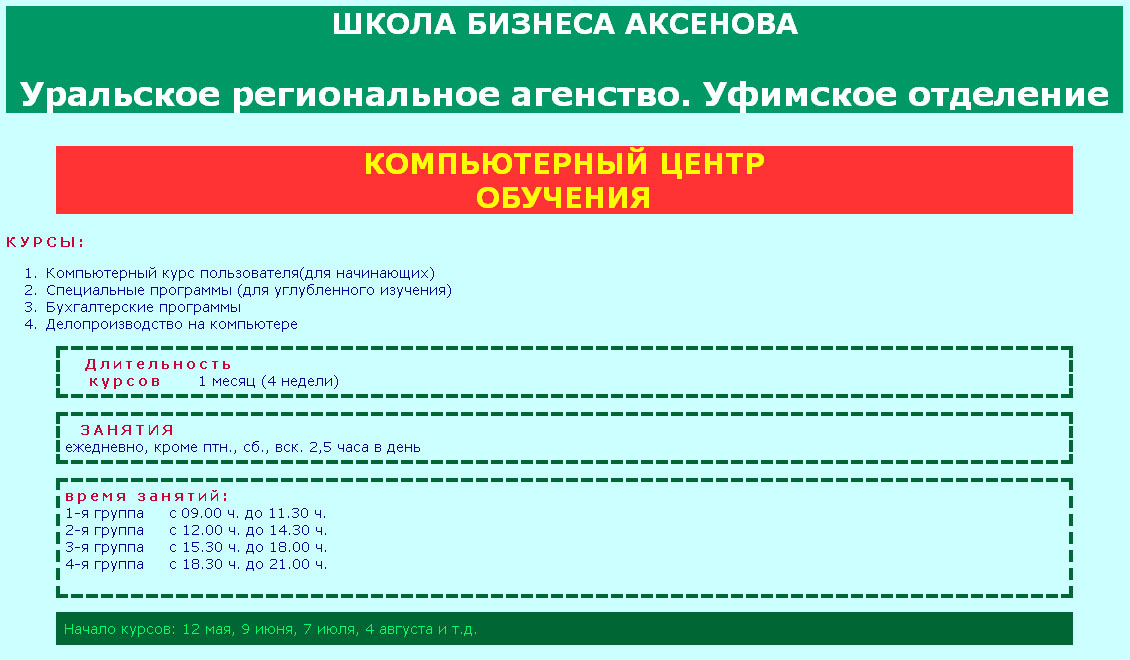
td.red21ine {border-color: #FF0000;

border-left: 2px double;

**}**

**Задание для практического занятия:**

1.Отформатируйте страницу с помощью листа стилей, размещённого в теге <style>...</style> (см. пример) Сохраните страницу в личной папке в файл style1.html



2.Запишите исходный код таблицы стиля в отчет

**Контрольные вопросы:**

1**.**Что является компонентами страницы HTML?

2.Какие существуют способы управления настройками браузера:

3.В чем главная особенность каскадных таблиц стилей?

**Задания для контрольной работы и требования,**

**предъявляемые к оформлению**

**контрольной работы**

В соответствии с учебными планами студенты специальности «Компьютерные системы и комплексы», должны выполнить контрольную работу по одному из 2 вариантов. Варианты устанавливаются по последней цифре в списке учебного журнала (нечетный номер – первый вариант, четный - второй).

Выполнение контрольного задания должно помочь студентам изучить операционные системы и среды проверить степень усвоения изученного материала и применить свои знания при решении лабораторных работ.

Перед выполнением контрольной работы студенту необходимо изучить рекомендованную литературу.

При выполнении контрольного задания следует руководствоваться методическими указаниями, где приведены примеры задач.

#### Требования к оформлению контрольной работы

Контрольная работа должна быть аккуратно оформлена и выполнена в распечатанном виде, на листе формата А4 или в ученической тетради в клетку темными чернилами (синими, черными, фиолетовыми) через строчку. Все дополнительные страницы должны быть в тетради приклеены или вшиты.

1. Контрольная работа начинается с титульного листа с указанием дисциплины и фамилией студента (образец 1)
2. Текст печатается шрифтом п. 12, Times New Roman, 1 интервал.
3. Поля страниц: верхнее - 1,5 см., нижнее - 2,0 см., левое – 3,0 см., правое – 1,0 см. Отступ абзаца 1,25 см.
4. Страницы следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту работы, не включая приложения. Номера страниц проставляют в середине нижнего поля. Титульный лист включается в общую нумерацию страниц, но номер страницы на них не проставляется. Таким образом, как правило, нумерация начинается с 2-ой страницы (раздел «Содержание»).
5. Подчёркивание, курсив, жирный шрифт в тексте не допускается. ЖИРНЫЙ ШРИФТ ДОПУСКАЕТСЯ **ТОЛЬКО** В НАЗВАНИЯХ ГЛАВ И ПАРАГРАФОВ! Заголовки глав, содержание, заключение, список использованной литературы следует располагать в середине строки без точки в конце и печатать ЗАГЛАВНЫМИ буквами.

Все страницы, формулы и таблицы нумеруются. Нумерация – сквозная (т.е. номер – один, два и т.д.).

Работа должна быть выполнена в той же последовательности, в какой приведены вопросы домашнего задания.

Следует полностью записывать формулировку вопроса согласно заданию, затем давать ответ.

Сокращение наименований и таблицы в задачах должны выполняться с учетом требований ЕСКД. При переносе таблиц следует повторить заголовок таблицы, указывая над ней «Продолжение таблицы» и ее номер. Единицы измерения указывать только в результирующих значениях.

В контрольной работе должны быть приведены условия задач, исходные данные и решения. Решение должно сопровождаться четкой постановкой вопроса (например, «Определяю …»); указываться используемые в расчетах формулы с пояснением буквенных обозначений; выполненные расчеты и полученные результаты должны быть пояснены.

Вычисление абсолютных величин следует производить с точностью до первого десятичного знака (0,1), в процентах – до первого десятичного знака (0,1%); относительных величинах – до второго десятичного знака (0,01).

В конце работы приводится список использованной литературы.

Титульный лист работы должен быть оформлен в соответствии с утвержденной формой, подписан, с указанием даты сдачи работы.

Например:

**ГЛАВА**

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Расстояние между заголовком глав и параграфом, как и между параграфом и последующим текстом, должно быть – пропуск 1 рабочей строки. Заголовки параграфов следует начинать с абзацного отступа (вторая строка заголовка пишется под первой буквой названия параграфа) и печатать с прописной буквы без точки в конце.

Например:

* 1. **Оптимизация объема производства и реализации в целях**

**увеличения финансовых результатов**

Главы и параграфы следуют нумеровать арабскими цифрами. При этом главы нумеруются одной цифрой с точкой. Например, 1., 2., 3. и т.д. ПЕРЕНОСЫ В ЗАГОЛОВКАХ НЕ ДОПУСКАЮТСЯ.

Например:

### ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПЛАТЕЖЕСПОСОБНОСТИ И ФИНАНСОВОЙ УСТОЙЧИВОСТИ ОРГАНИЗАЦИИ

Параграфы должны иметь порядковую нумерацию в пределах каждой главы и нумероваться двумя цифрами с точками, например, 1.1., 1.2., 2.2.,3.2. и т.д.

Например:

* 1. **Теоретические аспекты платежеспособности и финансовой устойчивости организации**

Если заголовок включает несколько предложений, их разделяют точками. Заголовки (названия глав и параграфов) должны включать от 2 до 14 слов (не более 2-3 строк).

Расстояние между последним предложением предыдущего параграфа и названием нового следующего параграфа составляет – пропуск 2 рабочих строки, с третьей начинается новый параграф.

Не разрешается размещать заголовки новых параграфов в нижней части страницы, если на ней помещается не более 4-5 строк текста нового параграфа.

Каждая глава должна начинаться с новой страницы.

При приведении цифрового материала должны использоваться только арабские цифры, за исключением общепринятой нумерации кварталов, полугодий, которые обозначаются римскими цифрами. Количественные числительные в тексте даются без падежных окончаний.

При величинах, имеющих два предела, единица измерения пишется только один раз при второй цифре. Такие знаки, как №, % пишутся только при цифровых или буквенных величинах. В тексте их следует писать только словами. Например, номер, процент и т.д. Математические знаки +, -, =, <, > и другие используются только в формулах. В тексте их следует писать словами. Например, плюс, минус, равно, меньше, больше.

### Оформление формул

Формулы располагаются либо по центру, либо с абзаца.

Значение символов и числовых коэффициентов должны быть приведены непосредственно под формулой, с новой строки в той же последовательности, в которой они приведены в формуле (причем, после формулы ставится запятая). Первая строка пояснения начинается словом «где» без двоеточия после него.

Нумеруются формулы арабскими цифрами в круглых скобках у правого края страницы. Нумерация единая и сквозная, начиная с первой главы и до конца работы (см. образец). Например, формула (1).

### Требования и правила выполнения графических работ

В контрольной работе могут быть различные графические работы, которые представляются в виде графиков, диаграмм, рисунков, схем. К выполнению графических работ также предъявляются определенные требования..

Все иллюстрации называются рисунками, которым присваивается последовательная нумерация – сквозная, единая, начиная с первой главы и до конца работы. Например, Рисунок 1., Рисунок 2. и т.д.

Все рисунки должны иметь подрисуночную подпись, которая располагается по центру. Причем, если вторая строка подписи рисунка состоит из одного слова, то она пишется по центру, в противном случае, под первой буквой названия. Ссылки на рисунок – (см. рис. 1). Точка после названия рисунка НЕ СТАВИТСЯ! Переносы в подрисуночной подписи НЕ ДОПУСКАЮТСЯ!

### Оформление таблиц

Таблицы последовательно нумеруются в пределах всего материала. Над правым верхним углом таблицы помещается надпись «Таблица» с указанием ее порядкового номера. Нумерация единая, сквозная, начиная с первой главы и до конца работы. Например, Таблица 12 - Расчет показаний вольтметра

Тематический заголовок определяет содержание таблицы и располагается по середине над таблицей (на следующей строке после номера таблицы). Точка после номера таблицы и в заголовке НЕ СТАВИТСЯ. Единицы измерения пишутся на следующей строке после названия таблицы в правом верхнем углу таблиц (предварительно в конце тематического заголовка ставится запятая). Жирный шрифт и подчеркивание в таблице не допускаются

На все таблицы должны быть ссылки в тексте, при этом слово «таблица» в тексте пишется полностью (например, в таблице 12). А также – (см. табл.12).

### Правила оформления списка использованных источников

Использованные в процессе работы над ВКР литературные источники указываются в конце работы перед приложением. Каждый литературный источник отражается в списке в следующем порядке:

* порядковый номер в списке;
* фамилия и инициалы автора;
* название книги (для статьи ее заглавие, название сборника, журнала, его номер);
* издательство;
* место и год выпуска;
* количество страниц в книге (256 с.); для статьи указывается номер страницы, где была помещена статья (с. 8).

Ссылка на литературный источник в тексте оформляется следующим образом: приводится порядковый номер использованной литературы и номер страницы, заключенные в квадратные скобки. Например, [1, с.2] - это значит первый источник из списка использованной литературы на второй странице.

### Оформление приложений

В приложении могут выноситься рисунки, графики, таблицы, диаграммы, формы бухгалтерской отчетности и т.д.

Приложения располагаются на последних страницах в виде самостоятельного раздела (см. образец). Каждое приложение должно начинаться с новой, страницы и иметь содержательный заголовок. В правом верхнем углу над заголовком должно быть слово «ПРИЛОЖЕНИЕ 1». Если одно приложение располагается на нескольких страницах (например, одна и та же таблица), то на всех этих страницах пишется слово «ПРИЛОЖЕНИЕ 1», причем, нумерация страниц продолжается. Начинается другое приложение – пишется «ПРИЛОЖЕНИЕ 2». Приложения нумеруются последовательно арабскими цифрами.

Приложения должны иметь общую с остальной частью работы сквозную нумерацию страниц.

***Образец 1 оформления титульного листа***

**Уфимский государственный колледж радиоэлектроники**

**КОНТРОЛЬНАЯ РАБОТА**

по дисциплине \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вариант\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил студент (ка) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О. студента)

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.

(дата сдачи)

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О. преподавателя)

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.

(дата проверки)

Уфа 2015

#### Контрольные задания

**Вариант 1**

3.1Создайте мини-сайт «Зоопарк» по вариантам

Разметку страниц сделайте с помощью таблиц:

Название сайта

Вторая таблица

Первая таблица

Содержание

страницы

Навигация

(список разделов)

**Страницы сайта:**

**1. Зоо-экскурсия: Гепард** (текст: **гепард.txt**, рисунки: **gepard.jpg**, **gepard-s.jpg**).

**2. Зоо-обои** (фото: **parrot.jpg**, **parrot-s.jpg**, **tiger.jpg**, **tiger-s.jpg**).

**Примерный вид страницы Зоо-экскурсия:**



Зоопарк.ру

# Гепард

Текст текст текст текст текст

**Гепард**

Зоо-обои

Почта:

info@zoopark.ru

баннер www.auto.ru.gif

**При создании сайта учитывайте следующие пожелания:**

1) Используйте теги заголовков (H1..H6) и параграфов (P).

2) Маленькие рисунки должны быть ссылками на большие рисунки.

3) Все имена файлов должны писаться *латинскими буквами* и/или цифрами.

4) Сделайте ссылку на почтовый ящик info@zoopark.ru

5) Поместите баннер на сайт www.auto.ru в левую ячейку таблицы.

**Вариант 2**

3.1Создайте мини-сайт «Москва» по вариантам

Разметку страниц сделайте с помощью таблиц:

Название сайта

Вторая таблица

Первая таблица

Баннеры

Ссылки

Навигация

(список разделов)

Содержание

страницы

**Страницы сайта:**

**1. Справка о Москве** (текст: **справка.txt**, рисунки: **v1.jpg**, **v2.jpg**, **v3.jpg**)

**2. Московские обои** (рисунки: **moskva-1.jpg**, **moskva-1-s.jpg**, **moskva-2.jpg**, **moskva-2-s.jpg**)



**Примерный вид страницы Справка о Москве:**

Москва.ру

**Справка о Москве**

Московские обои

Почта:

info@moskva.ru

Справка о Москве

Текст текст

баннер www.auto.ru.gif

**При создании сайта учитывайте следующие пожелания:**

1) Используйте теги заголовков (H1..H6) и параграфов (P).

2) Маленькие рисунки должны быть ссылками на большие рисунки.

3) Все имена файлов должны писаться *латинскими буквами* и/или цифрами.

4) Сделайте ссылку на почтовый ящик info@moskva.ru

5) Поместите баннер на сайт www.new.auto.ru в правую ячейку таблицы.

**Литература**

1. Кирсанов Д. Web-дизайн: книга Дмитрия Кирсанова. – СПб: Символ-Плюс, 2010

2. Компьютерная графика в дизайне: Учебник для вузов/Д.Ф. Миронов.– СПб.: Питер, 2010

3. Клонингер К. Свежие стили Web-дизайна: как сделать из вашего сайта «конфетку»: Пер с англ. – М.: ДМК Пресс, 2011

4. Левин А.Ш. Самоучитель компьютерной графики и звука. – СПб.: Издательство «Питер», 2010

5. Мак-Клелланд, Дик, Обермайер, Барбара Photoshop CS5 для «чайников» .: Пер. с англ. – М.: Издательский дом «Вильямс», 2010

6. Немцова Т.И., Назарова Ю.В. Информатика. практикум по информатике: учеб. Пособие/ Под ред. Л.Г. Гагариной Ч. I. – М.: ИД «ФОРУМ»: ИНФРА-М, 2010

7. Немцова Т.И., Назарова Ю.В. Комппьютеная графика и web-дизайн практикум по информатике: учеб. Пособие/ Под ред. Л.Г. Гагариной – М.: ИД «ФОРУМ»: ИНФРА-М, 2010

8. Храмцов П.Б. Основы Web-технологий: учебное пособие / П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин – 2-е изд., испр. –М: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2010

Дополнительные источники:

1. Шапошников И. Web‑сайт своими руками. – Спб: Издательство «БХВ – Санкт Петербург», 2009

2. Шарф Д. HTML 3.2: справочник. - СПб.: Издательство «Питер», 2009

Интернет ресурсы:

1. Журнал веб-дизайн – уголок профессионала. [Электронный ресурс] – режим доступа: <http://www.webmagazine.biz/> (2005-2015)

2. Онлайн-журнал для профессиональных веб-дизайнеров и разработчиков. [Электронный ресурс] – режим доступа: <http://www.coolwebmasters.com/> (2009-2015)

3. Система федеральных образовательных порталов Информационно -коммуникационные технологии в образовании. [Электронный ресурс] – режим доступа: <http://www.ict.edu.ru> (2003-2015)