# ПРАКТИКУМ: ОСНОВЫ HTML И CSS В ВЕБ-ДИЗАЙНЕ

Веб-страница представляет собой электронный документ, содержимое которого включает как непосредственно контент страницы, так и правила отображения этого контента, указывающие программам просмотра (браузерам) форму представления описанной в документе информации. Эти правила составляются с помощью языка гипертекстовой разметки HTML (Hyper Text Markup Language).

Разметка – это код, который сопровождает содержание документа   
и предоставляет браузеру (или другой обрабатывающей программе) информацию о том, как следует интерпретировать те или иные фрагменты содержания.

Язык HTML предназначен в первую очередь для структурной (логической) разметки, т. е. для обозначения каждого элемента в соответствии с его местом и ролью в структуре документа. Структурная разметка не имеет никакого отношения к внешнему виду документа. Документ может быть отображен   
на экране компьютера, написан от руки или напечатан на пишущей машинке, но все заголовки в нем останутся заголовками, все пeречни –перечнями, а цитаты –цитатами. Структурная роль элементов документа, в отличие от оформления, никак не меняется в зависимости от того, какими средствами просматривают этот документ.

Таким образом, [*структурная разметка*](http://www.seo-copywrite.ru/51/) – это разметка, обозначающая роль каждого элемента в структуре документа. Структурная разметка говорит   
о том, чем является тот или иной элемент, а не о том, как его следует или   
не следует отображать. Грамотная структурная разметка обеспечивает независимость документа от устройства вывода.

Структурная разметка не говорит о том, как конкретно должен быть выделен текст. Но это не мешает разработчику оформить его нужным образом.

Оформительские возможности HTML очень скудны. Для эффектного оформления отдельных элементов и придания комплексу веб-страниц единого узнаваемого стиля используют технологию CSS (Cascading Style Sheets – [каскадные таблицы стилей](http://www.seo-copywrite.ru/css2/)). Таким образом, структурная разметка и оформление дополняют друг друга.

Итак, на сегодняшний день язык HTML является основой всех разме­щенных в Интернете электронных документов. Он выступает в роли фундамента, на базе которого реализуются прочие сетевые программные технологии, призванные повысить общую привлекательность, эффективность и интерактивность носителей информационных данных. В этой связи веб-мастеру необходимо знание основ языковых средств создания веб-документов, умение использовать эти знания для создания собственных и для анализа чужих решений.

Далее в главе кратко изложены необходимые веб-дизайнеру основы HTML и CSS, оформленные в виде комплекса практических работ, последовательное выполнение которых обеспечит базовую подготовку специалиста к соз­данию и редактированию веб-сайтов.

## 3.1. Основные разделы HTML-документа. Описание раздела body

Язык HTML представляет собой набор определенных правил, в соответствии с которыми браузер отображает содержимое электронной страницы. Каждому правилу соответствует свое название (символьное обозначение, код), свойство (или набор свойств, называемых параметрами либо атрибутами)   
и значение (значения этих атрибутов). Для выделения управляющего кода   
из текстового массива его символьное обозначение заключается в угловые скобки – <…>.

Подобная конструкция <набор текстовых символов> называется *тегом*. Язык HTML не чувствителен к регистру текстовых символов, поэтому название тегов и атрибутов допускается набирать как прописными (<BODY>), так   
и строчными (<body>) символами.

Теги бывают одинарными и парными:

* пример непарного тега: <hr>;
* пример парного тега: <b>….</b>.

В структуре HTML-документа можно выделить три части.

1. Строка, содержащая декларацию типа документа;
2. Заголовок (шапка) документа;
3. Тело документа.

В первой строке заявляется тип документа (DOCTYPE), а также правила, которым следует разметка и согласно которым она будет проверяться. Правила оформления строки можно найти на соответствующих ресурсах сети Интернет. Основные правила представлены в прил. 1.

Вторая и третья части находятся в границах тега HTML. Таким образом, схема HTML-документа выглядит так:

<html>

<head>

Раздел заголовка

Раздел тела документа

……….

</head>

<body>

………..

</body>

</html>

В *разделе заголовка,* ограниченного контейнером <head> … </head>, могут размещаться определённые теги, которые не вносят изменений в видимое пользователю содержимое веб-страницы, но влияют на связь между документами и оформление гиперссылок. Для оформления заголовка окна документа используется <title> … </title>. Данный тег-контейнер задает название страницы, которое отображается в строке заголовка окна браузера.

В *разделе тела документа*, ограниченного контейнером <body> … </body> (или <frameset>…</frameset> для страниц, созданных на основе фреймов), располагается вся содержательная часть веб-страницы, которая выводится браузером на экран монитора пользователя. Внутри этого раздела может размещаться большинство существующих тегов HTML.

Каждый тег может иметь один или несколько атрибутов.

*Атрибуты* – имена свойств тегов, которые могут принимать определенные значения. Атрибуты имеются только у открывающих тегов, у закрывающих тегов их нет. Атрибуты отделяются от тега и других атрибутов пробелами. Если атрибут имеет конкретное значение, то оно задается после знака равенст­ва. Атрибуты влияют на результат интерпретации тега браузером. Регистр написания тегов и атрибутов значения не имеет.

Оформление конструкции тега, содержащего атрибут, выглядит следующим образом:

Имя атрибута

Значение атрибута

Имя тега

<h3 align=” center”> Страница должна иметь заголовок </h3>

Закрывающий тег

Открывающий тег

Описание тела документа.

В теге <body> задаются свойства документа. Для этого используют ат­рибуты:

bgcolor – задание цвета заднего фона;

background – задание фонового рисунка;

bgproperties – изменение свойств заднего фона;

topmargin,

bottommargin,

leftmargin, – определение размеров отступов;

rightmargin,

marginwidth,

marginheight

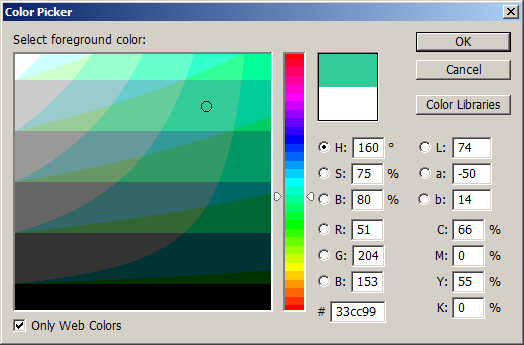
text – задание цвета основного текста;

link, alink, vlink – задание цвета гиперссылок.

Атрибут bgcolor задает однородный цвет фона документа, значение которого может быть введено в символьном виде или в шестнадцатеричном коде:

<body bgcolor=“red”> или <body bgcolor=“#ff0000”>.

При определении цвета элементов HTML-страницы рекомендуется использовать так называемые безопасные цвета. Безопасные цвета веб-палитры можно выбрать, например, в окне сборщика цветов программы Adobe Photoshop (рис. 40).



Номер выбранного цвета   
в шестнадцатеричном коде

При установленной галочке в окне выбора цвета отображаются только цвета Web-палитры

*Рис. 40.* Палитра выбора цвета в Adobe Photoshop

При использовании background на задний фон документа помещается графическое изображение, которое, как обои с определенным рапортом, заполнит фон всего документа:

<body background=“images/logo.gif”>.

Существуют определенные требования к изображениям, помещаемым на HTML-страницы. Файл с изображением должен быть в цветовой модели RGB или Index и сохранен в формате jpeg или gif. Физический размер изображения (размер растровой сетки в пикселах) и размер файла с изображением должны быть приемлемыми для веб-размещения.

bgproperties – задает свойство фонового изображения. При значении fixed запрещает прокрутку фонового изображения совместно с текстом страницы. Например:

<body background=“images/logo.gif” bgproperties=“fixed”>.

text – задает используемый по умолчанию цвет текста страницы, не являющегося гиперссылкой (если атрибут не используется, то цвет по умолчанию – черный). Цвет шрифта может быть задан:

<body text =“blue”> или <body text =“#00ffff”> .

link, alink, vlink – задает цвета: не открытых гиперссылок страницы, активной гиперссылки и посещенной соответственно. Например, если все не посещенные ссылки требуется отображать синим, активную – красным, а посещенную пурпурным цветами, то можно задать конструкцию

<body link=”#0000ff” alink=”#ff0000” vlink =”#ff00ff”>.

Атрибуты topmargin, bottommargin, leftmargin, rightmargin задают: верхний, нижний, левый и правый отступы заданного размера от краев документа. Значения отступов указываются в пикселях:

<body topmargin=“10” leftmargin=“15” rightmargin=“15”>.

Если для описания тела документа необходимо указать несколько атрибутов одновременно, то запись может выглядеть следующим образом:

<body background=“images/logo.gif” bgproperties=“fixed” link=”#0000ff” alink=”#ff0000” vlink =“blue” topmargin=“10” leftmargin=“15” rightmargin=“15”>.

Дополнительную информацию о структуре HTML-документа и описании его свойств в теге <body> можно найти на соответствующих ресурсах сети Интернет, например на образовательных ресурсах: <http://postroika.ru>; <http://intuit.ru> и пр.

Практическая работа

Основные разделы HTML-документа. Описание раздела body

1. Создайте в текстовом редакторе (Блокнот) шаблон веб-страницы, сохраните его как HTML-документ с именем listing. Для этого в окне сохранения документа в поле «Имя файла» введите строчными латинскими буквами без пробелов имя файла и после точки расширение: html или htm.

Обратите внимание: созданный HTML-документ можно просматривать как с помощью текстового редактора (в виде текста на языке HTML), так и через браузер – как веб-страницу.

2. Создайте веб-страницу, для которой заданы: верхняя, левая и правая границы по 15 пикс., на заднем плане – фоновое изображение, цвет текста – отличный от черного. Сохраните изменения.

3. Откройте сохраненный документ listing.html в редакторе Notepad++ (программа предоставляется для свободного использования на официальном сайте <http://notepad-plus-plus.org)//>. Обратите внимание на разницу в визуальном представлении текста html-кода. Сделайте вывод об удобстве использования того или иного текстового редактора для записи HTML-кода гипертекстового документа. В дальнейшей работе используйте выбранный редактор.

4. Скопируйте из любого текстового документа в раздел тела HTML-документа значительное количество текста, которое, при просмотре страницы   
в браузере не будет умещаться в одном (первом) экране.

5. Оцените внешний вид полученного документа. Обратите внимание   
на появление полосы прокрутки и на то, как выглядит документ при прокру­чивании. Сделайте вывод об удобстве чтения текстовой информации с экрана   
и о том, как фон и цвет текста влияют на удобочитаемость.

6. Добавьте в код документа атрибут, фиксирующий фоновое изображение. Убедитесь в том, что при прокручивании текста полосой прокрутки изображение заднего фона остается неподвижным.

7. Откройте созданный HTML-документ в различных браузерах (Internet Explorer, Opera, Safari и др). Сравните вид открытого документа, определите различия, сделайте выводы.

8. Сохраните файл на рабочем столе учебного компьютера для предъявления преподавателю и, при отсутствии замечаний к результатам работы, удалите файл после проверки.

Форма отчета

1. Результатом выполнения работы является содержимое HTML-документа listing.html, которое демонстрируется: 1) в окне текстового редактора; 2) в окне браузера.
2. Содержимое созданного документа должно включать: задание верхней, левой и правой границ документа по 15 пикс. с каждой стороны; установку цвета основного текста, отличного от цвета по умолчанию; установку адекватного (по размерам и цветовому оформлению) изображения на задний фон HTML-страницы, фиксацию изображения заднего фона.

Контрольные вопросы

1. Что представляет собой язык HTML? Что называют тегом, атрибутом тега?
2. Какова общая структура HTML-документов? Назовите назначение разделов HTML-документа.
3. Какие свойства документа задаются при описании <body>? Назовите основные атрибуты раздела тела документа и атрибуты тега <body>.
4. Назовите способы задания цвета элементов HTML-документа. Какие цвета называют безопасными?
5. Назовите условия, которым должны удовлетворять графические изображения, предназначенные для размещения на заднем плане (фон) веб-страницы.

**Домашняя работа**

**Анализ заднего фона веб-страниц сайтов различных категорий.   
Создание фонового изображения для веб-страницы**

Цель работы

Получить представление об особенностях выбора и создания фонового изображения веб-страниц, для чего:

* провести анализ заднего фона веб-страниц сайтов различных категорий;
* на основе результатов проведенного анализа создать собственное изображение для заднего фона веб-страницы с учетом предполагаемой тематики будущего сайта и различия параметров разрешения мониторов.

Краткие теоретические сведения

Веб-сайт представляет собой совокупность электронных страниц, объединенных по смыслу и стилевому оформлению, связанных системой гиперссылок и физически находящихся на одном сервере. Целостный образ сайта соз­дается сочетанием контента (содержательной части, наполнения) и оформления его страниц.

Для улучшения эстетического вида и узнаваемости веб-страницы, а также для облегчения восприятия информации на ней, используется задний фон (background). Фоном называют графическое изображение, размещенное на заднем плане страницы. Фоновое изображение предназначается для облегчения восприятия информации на странице, а также создания определенного визуального образа. Выбор фонового изображения задает визуальный образ страницы   
и определяется тематикой (направленностью) ресурса.

В Интернет присутствуют:

* сайты с деловой информацией (сайты биржевых и бизнес-компаний; промышленных, производственных корпораций и др., задачей которых является оперативное и полное представление достоверной информации);
* рекламно-информационные сайты и порталы (сайты производителей   
  и продавцов товаров и услуг массового потребления, правительственные, образовательные, общественные сайты и пр., задачей которых является информирование об определенных сферах общественной жизни);
* развлекательные, художественные сайты (сайты досуговых и развлекательных заведений, электронные выставочные площадки и видеозалы, персональные страницы, являющиеся продуктами самовыражения творческой личности и прочие ресурсы, не несущие серьезной информационной нагрузки, задачей которых является развлечение и неутомительное просвещение посетителя) и др.

Каждой категории сайтов (информационных ресурсов) присущи определенные традиции в оформлении заднего фона страниц, вызванные объективными причинами: задачами сайта, образом компании, предпочтениями целевой аудитории и пр.

Указания к выполнению домашней работы

1. Найдите в сети Интернет страницы веб-сайтов различного назначения (не менее трех примеров сайтов каждой категории). Обратите внимание   
   на оформление фона веб-страниц. Сделайте вывод об особенностях оформления заднего плана страниц для сайтов единой категории.
2. На основе проведенных наблюдений и анализа фоновых изображений рекламно-информационных сайтов продумайте внешний вид и выполните   
   на бумаге наброски фона воображаемого сайта задуманной тематики.
3. Используя графический редактор Adobe Photoshop, создайте два изображения, предназначаемых для помещения на задний фон веб-страницы:

3.1) рисунок фона (фрагмент[[1]](#footnote-1)), который будет тиражироваться слева направо и сверху вниз, заполняя все пространство окна браузера;

3.2) рисунок, композиция которого предусматривает корректное покрытие заднего плана веб-страницы без тиражирования (клонирования) при открытии ее (страницы) на экране монитора с размерами от 800×600 до 1920×1080 пикселей.

Форма отчета

1. Результат выполнения пунктов 1 и 2 данной работы представляется   
   в форме электронного текстового документа, содержащего:

* для каждой группы (категории) сайтов: наименование группы; описание задач, на выполнение которых нацелены сайты этой группы; описание факторов, определяющих предпочтения в оформлении заднего плана веб-страниц; перечень сайтов, рассмотренных в данной группе (с указанием URL-адресов найденных ресурсов); скриншот[[2]](#footnote-2) характерного примера страницы сайта данной категории;
* вывод, сделанный по результатам наблюдений, о замеченных предпочтениях при выборе фона страниц для сайтов каждой категории.

1. Результат выполнения п. 3 Указаний к выполнению домашней работы (далее – Указаний) представляется комплектом файлов, демонстрируюших:

* образец текстурной (фактурной) заливки веб-страницы (файл формата jpg, gif или png), созданный при выполнении п. 3.1;
* полноформатный рисунок фона веб-страницы (файл формата jpg, gif или png), созданный при выполнении п. 3.2;
* вид веб-страницы[[3]](#footnote-3), на задний план которой установлен рисунок, созданный при выполнении п. 3.1 или 3.2 (файлы формата html).

Контрольные вопросы

1. Назовите особенности визуального оформления заднего плана веб-страниц для сайтов различной направленности.
2. Какие требования предъявляются к рисункам текстуры, автоматически заполняющим задний план веб-страницы, к рисункам, устанавливаемым на задний план страницы без тиражирования?

## 3.2. Форматирование текста в HTML

В большинстве случаев посетители веб-страниц обращаются на сайт   
в поисках текстовой информации. Текст является важной составляющей HTML-документа и от того, как он преподнесен читателю, зависит удобство работы с сайтом и в конечном итоге – его успех.

Под форматированием текста понимается возможность с помощью специальных тегов красиво располагать текст на странице, выводить в виде всплывающих подсказок, расставлять смысловые акценты применением различного шрифта, цвета, отступов и т. п.

**Структурное форматирование документа**

Любой документ, созданный с определенной логикой, имеет свою структуру. Структура учебной книги, например, предполагает разделение на главы, параграфы, пункты, которые снабжены заголовками. Кроме того, в ней могут быть выделены практикумы, справочные разделы, вопросы для самоконтроля. Структурное форматирование электронного документа подразумевает разбиение текстовых фрагментов на логические блоки и включает: создание заголовков, установку абзацев, центрирование, отступы, перенос строки, вставку горизонтального разделителя, списков, добавление комментариев и т. п., то есть все то, что облегчает восприятие страницы пользователем и обработку документа автоматическими средствами.

*Заголовки*

Заголовки – важная составляющая контента веб-страницы: они помогают систематизировать текст. В HTML доступно создание заголовков разных уровней, поэтому легко выделять смысловые разделы и подразделы. При индексации текста страницы поисковыми системами особое внимание уделяется содержимому заголовков. Текст, оформленный как заголовок, имеет для поисковых роботов более высокий ранг.

В HTML-документе можно создать заголовки шести уровней.

<h1>, <h2>, … <h6> – используются для обозначения заголовков соответствующего уровня (от 1 до 6). Применение тега <h> для оформления заголовка подразумевает отступ от основной текстовой части и задание высоты букв   
заголовка, соответствующей его уровню (символы заголовка, размещенные   
в контейнере <h1> … </h1>, будут наибольшими по величине, а в <h6> … </h6> будут отображаться с минимальной высотой букв).

*Абзац*

В HTML-документе невозможно разбить текст на абзацы, используя клавишу Enter. Абзацы задаются использованием тега <p>.

Каждому заголовку или абзацу с помощью атрибута align может быть задан определенный тип выравнивания.

align – атрибут выравнивания. Может принимать значения:

left – выравнивание по левому краю (значение по умолчанию);

center – выравнивание по центру;

right – выравнивание по правому краю;

justify – выравнивание по ширине окна браузера.

*Обрыв строки*

При необходимости оборвать строку, не закрывая абзаца, или же для принудительного помещения следующего элемента в начало новой строки применяется непарный тег <br>.

*Центрирование*

Центрирование любых элементов HTML-документа может быть осуществлено с помощью тега-контейнера <center>. Все данные, размещенные внутри контейнера <center> … </center>, подлежат выравниванию по горизонтальному центру окна браузера.

*Отступ*

В текстовом редакторе отступ задается клавишей табуляции Tab.   
В HTML-документе размер дополнительного отступа определяется кодовой конструкцией – символом пробела – &nbsp;. Если необходимо сделать отступ, соответствующий визуально трем пробелам, то для обеспечения смещения потребуется трижды повторить указанный символ: &nbsp; &nbsp; &nbsp;.

*Горизонтальный разделитель*

Для разделения фрагментов текста в соответствии с логикой подачи материала на странице сайта веб-дизайнеры проводят визуальную границу между текстовыми блоками, так называемый *горизонтальный* *разделитель*.

<hr> – тег создания горизонтальной разделительной линии.

Тег hr может включать ряд атрибутов, описывающих свойства разделительной линии:

width – указание длины линии. Значение может задаваться абсолютно (количеством пикселей) или относительно (в процентах от ширины окна браузера).

size – указание толщины линии. Значение толщины указывается в пик­селях;

align – выбор типа выравнивания линии. Типы выравнивания: left, center, right;

color – выбор цвета линии;

noshade – отмена эффекта рельефности линии.

Конструкция для задания горизонтальной линии зеленого цвета, толщиной 3 пикселя, растянутой на 80 % от ширины окна браузера и выровненной по правому краю, будет иметь вид

<hr color=”green” size=”3” width=”80%” align =” right”>.

*Предварительно отформатированный текст*

Тег-контейнер <pre> … </pre> позволяет выводить на экран монитора текст в таком виде, как он выглядит в окне текстового редактора: со всеми переносами строк, отступами, абзацами и прочим форматированием.

*Списки*

Списки – важный инструмент, применяемый для организации и группировки данных. В HTML можно выделить списки нескольких типов:

* маркированный – список, пункты которого отмечены с помощью различных символов (применяется, когда порядок следования пунктов не имеет принципиального значения);
* нумерованный – список, пункты которого упорядочены и последовательно обозначены цифрами или буквами алфавита (применяется, когда порядок следования пунктов имеет большое значение);
* список определений – вид списка, который применяется для формирования словарей или когда необходимо пояснять значения терминов.

Для задания списков используются теги-контейнеры:

<ul> – задает маркированный список;

<ol> – задает нумерованный список;

<dl> – задает список определений.

Для тегов <ul> и <ol> может задаваться атрибут type, определяющий вид маркера или вид нумерации. Значения атрибута type для маркированных и нумерованных списков приведены ниже.

|  |  |  |
| --- | --- | --- |
| Тип списка | Значение  атрибута type | Вид создаваемого маркера |
| Маркированный | сircle  square  disc | маркер в виде окружности;  маркер в виде квадрата;  маркер в виде черного круга |
| Нумерованный | A  a  I  i  1 | заглавные буквы латинского алфавита;  строчные буквы латинского алфавита;  заглавные римские цифры;  строчные римские цифры;  арабские цифры |

Для нумерованного списка существует возможность с помощью атрибута start указать, с какого порядкового номера начать нумерацию всего списка. Например, при использовании конструкции (<ol type=”a” start=”4”>, нумерация будет производиться заглавными строчными буквами и начнется с «D».

Для задания элемента маркированного или нумерованного списка используют тег li. HTML-код оформления списка выглядит следующим образом:

<ol type=”a” start=”4”>

<li> Текст пункта первого;

<li> Текст пункта второго;

<li> Текст пункта третьего.

</ol>

Для задания элементов списка определений вместо тега LI используют комплект парных тегов:

<dt> – для задания термина;

<dd> – для задания определения термина.

HTML-код списка определений выглядит следующим образом:

<dl>

<dt> Термин 1 </dt>

<dd> Определение термина 1 </dd>

<dt> Термин 2 </dt>

<dd> Определение термина 2 </dd>

</dl>

*Комментарии*

Разработчику веб-страницы, для того чтобы легко ориентироваться в записи HTML-кода документа, может потребоваться добавление в структуру кодовой страницы неотображаемых в окне браузера пояснений. Такие не видимые пользователю при просмотре в браузере пояснения называют комментариями.

Любой комментарий начинается с символов <!-- и заканчивается -->. Между ними находится текст комментария, который может содержать буквы, цифры, символы и прочее, за исключением тегов. В комментарии можно указывать информацию об авторском праве, персональные данные, названия структурных блоков страницы и прочее.

<!-- Начало структурного блока -->

Код заявленного блока

<!-- Конец структурного блока -->

Указанные в комментариях данные будут проиндексированы поисковыми системами.

Физическое форматирование текста

К физическому форматированию относится возможность визуально изменять текст, варьируя его параметры (цвет, гарнитуру, размер и т. п.) произвольным образом. Теги физического форматирования предназначены для выделения отдельных текстовых фрагментов различными способами по усмотрению автора документа.

Тег <font> – один из основных тегов физического форматирования. С его помощью задаются такие свойства шрифта, как гарнитура, цвет, размер.   
Для этого внутри font могут использоваться следующие атрибуты:

face – задает тип шрифта, которым должен отображаться заключенный   
в контейнер текст веб-страницы. Если на компьютере пользователя не установлен шрифт, указанный в HTML-коде, то текст будет показан стандартным   
(по умолчанию) шрифтом. Допускается указывать сразу несколько значений параметра FACE, чтобы в случае отсутствия на компьютере первого шрифта браузер отобразил текст вторым шрифтом из списка. Например:

<font face=”tahoma”,”arial”>текст, к которому применяется оформление</font>.

color – задает цвет текста, которым будет написан фрагмент, заключенный в контейнер.

size – управляет размером шрифта. Может принимать значения от 1 до 7. Размер шрифта можно указать абсолютным значением (задать число от 1 до 7) или относительным, то есть указать, на сколько пунктов больше или меньше текущего должен быть размер шрифта (например, +3 или –2). По умолчанию устанавливается величина шрифта номер 3.

Для физического форматирования применяются следующие теги:

<basefont> – определение свойств шрифта для всего документа;

<font> – задание свойств шрифта (гарнитура, размер, цвет);

<b>, <strong> – выделение жирным начертанием;

<i>, <em> – выделение курсивом;

<u> – выделение подчеркиванием;

<tt> – имитация телетайпа или моноширинного шрифта;

<big> – выделение фрагмента текста шрифтом большего размера, чем основной текст;

<small> – выделение фрагмента текста шрифтом большего размера, чем основной текст;

<span> – присвоение фрагменту текста определенных свойств в случае, когда невозможно использование тегов логического форматирования;

<s>, <strike> – обозначение удаленной информации зачеркиванием текста;

<sub> – оформление нижнего индекса;

<sup> – оформление верхнего индекса.

Логическое форматирование текста

Логическая разметка предназначена, в первую очередь, для обозначения каждого элемента в соответствии с его местом и ролью в структуре документа. Логическое форматирование предполагает отображение элементов веб-страницы на экране монитора таким образом, как установлено по умолчанию для элементов данного типа (например, заголовков или цитат) в специфике языка HTML.

Логическое форматирование может осуществляться с помощью тегов:

<abbr> – определяет текст как аббревиатуру;

<acronym> – определяет текст как акроним[[4]](#footnote-4);

<code> – указание фрагмента программного кода;

<del> – обозначение удаленной информации;

<em> – выделение курсивом;

<strong> – выделение жирным начертанием;

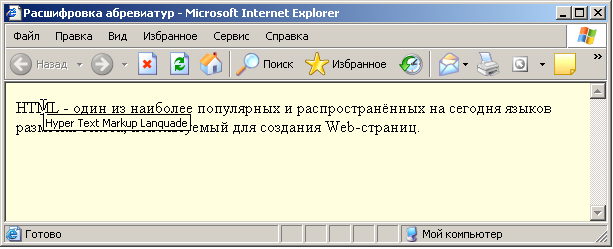
<cite> – выделяет специальным образом (курсивом) цитаты и высказывания, названия библиографических источников.

Теги <abbr> и <acronym> – позволяют создать всплывающие подсказки   
с помощью атрибута title для указания текста всплывающего пояснения. Результат использования конструкции:

<acronym title="Hyper Text Markup Language">HTML</acronym> – один   
из наиболее популярных и распространённых на сегодня языков разметки текс­та, используемый для создания web-страниц.

приведен на рис. 41.

При использовании тега <abbr> поисковые роботы индексируют именно полный вариант расшифровки, определенный атрибутом title.



*Рис. 41.* Результат использования тега <acronym>

Дополнительную информацию тегах структурного, физического и логического форматирования текста можно найти на соответствующих ресурсах сети Интернет.

Практическая работа

Форматирование текста в HTML

1. Создайте новый или откройте созданный ранее документ HTML. Опробуйте в действии команды физического, логического и структурного форматирования.

2. Спроектируйте вид веб-страницы, представляющей собой словарь терминов предметной области «веб-дизайн», используя подготовленные текстовые и графический материалы.

3. Применяя команды форматирования, создайте заготовку веб-страницы «Глоссарий[[5]](#footnote-5)», наполнение которой будет осуществлено в ходе следующей практической работы (прил. 2).

Форма отчета

Результат представляется в виде HTML-файла (или комплекта файлов, если на веб-странице установлено фоновое изображение), содержимое которого демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.

**Контрольные вопросы**

1. Что представляет собой текстовый дизайн сайта?
2. В чем заключается структурное форматирование текста? Расскажите   
   об использовании тегов <p> и < hr >.
3. В чем заключается физическое и логическое форматирование текста? Расскажите об использовании тегов физического и логического форматирования.
4. Расскажите о правилах оформления маркированных и нумерованных списков.

## 3.3. Организация гиперссылок

Ценность и популярность веб-сайтов во многом обусловлены наличием гиперсвязей – организованных возможностей быстрого перехода как на внутренние страницы текущего сайта, так и на внешние ресурсы сети. Гиперсвязь обеспечивается формированием гиперссылок.

Особенностью гиперссылки является то, что щелчок мыши по ней приводит к загрузке электронного документа, связанного с исходным при помощи электронного кода. В качестве гиперссылок могут выступать часть текста, изображение или фрагмент изображения, прочие элементы веб-страницы.

На веб-странице можно организовать переходы трех типов:

* *внутренние*, иначе – *якорные* (обеспечивают переход в пределах страницы для документов, размеры которых выходят за пределы одного экрана);
* *локальные* (обеспечивают переход на другие страницы того же самого сайта);
* *внешние*, иначе – *глобальные* (обеспечивают переход на внешние ресурсы сети).

Любая ссылка состоит из указателя ссылки и адреса ресурса, на который необходимо осуществить переход. В качестве указателя ссылки может выступать отдельное слово, фрагмент текста или графическое изображение.

Указатель ссылки описывается парным тегом <a>.

Внутри контейнера <a> … </a> размещают тот текст (рисунок), который на веб-странице должен быть выделен подчеркиванием. При наведении на него курсором указатель будет принимать вид руки с указательным пальцем, свидетельствующей о возможности перехода. Адрес перехода задается с помощью обязательного параметра href. Наиболее просто оформляются локальная и глобальная ссылки.

*Правила описания локальных и глобальных гиперссылок*

Для организации перехода на другую страницу или другой ресурс сети Интернет применяется конструкция

<a href=”Адрес перехода”>Текст, который является ссылкой </a>

href – указатель адреса перехода по гиперссылке. Значением этого атрибута может быть абсолютный адрес – полный путь к ресурсу или относительный – указание директории размещения требуемого документа относительно текущего.

Пример ссылки с указанием абсолютного адреса (для организации глобальной ссылки):

<a href=”http:/www.omgtu.ru”> Сайт ОмГТУ </a>

Пример ссылки с указанием относительного адреса (для организации локальной ссылки):

<a href=”images/dramteatr.jpg”>Омский театр драмы</a>

Тег <a> , помимо обязательного href, может содержать атрибуты:

name – именной идентификатор внутренней ссылки;

title – атрибут создания всплывающей подсказки;

target – атрибут задания окна, в котором браузеру следует открывать указанный в ссылке ресурс. Чтобы не уводить пользователя со страницы, а открывать ссылку в новом окне, параметру задается значение “\_blank”.

*Правила описания внутренних гиперссылок*

Если HTML-документ слишком большого размера и нет возможности разбить его на несколько отдельных файлов, можно предусмотреть переходы, перемещающие пользователя в пределах одной веб-страницы, избавляя его   
от необходимости прокрутки документа.

Структура внутренней ссылки включает в себя две части:

* саму ссылку,
* именной идентификатор.

Для оформления первой части (самой ссылки), то есть выражения или иного элемента, который будет выделяться как гиперссылка, предусматривается следующая конструкция:

<a href=”#якорь”> в начало </a>

Для обозначения места на странице, на которое нужно перенести пользователя документа со ссылки с указанным в якоре именем, используется конст­рукция

<a name=”якорь”> </a> … здесь начинается текст, в начало которого был оформлен переход по ссылке.

Если в качестве внутренней ссылки будет выступать рисунок (например, изображение стрелочки «вверх» из файла с именем top.gif), то первая часть ссылки будет иметь вид

<a href=”#якорь”> <img src=“images/top.gif”></a>

Дополнительную информацию об описании гиперссылок можно найти   
на соответствующих ресурсах сети Интернет, а также в учебной литературе.

**Практическая работа**

**Организация гиперссылок**

1. Откройте документ «Глоссарий», созданный в ходе предыдущей практической работы. В список терминов включите: веб-дизайн; веб-страница; веб-сервис; сайт и др.

2. Используя ресурсы сети Интернет, отыщите по 2–3 различных толкования каждого термина. Перенесите в созданный HTML-документ найденное определение термина с указанием источника цитирования. Посредством глобальной ссылки оформите гипертекстовый переход к странице, с которой было осуществлено копирование. Выполните эту операцию для каждого процитированного определения.

3. Оформите якорные ссылки для осуществления быстрых переходов:   
1) от термина, стоящего в списке, к участку страницы, содержащему его толкования; 2) снизу вверх, к списку терминов в начале веб-страницы.

Возможный вид страницы глоссария терминов предметной области «Веб-дизайн» представлен в прил. 2.

Форма отчета

Результат представляется в виде папки с файлами, включающей основной HTML-документ и все связанные с ним файлы. Содержимое основного HTML-документа демонстрируется преподавателю: 1) в окне текстового редактора;   
2) в окне браузера.

**Контрольные вопросы**

1. Что называется гиперссылкой? Что представляют собой внутренние, локальные и глобальные ссылки?
2. Расскажите об организации внутренней и локальной гиперссылок.

## 

## **3.4. Добавление графики на веб-страницу. Особенности веб-графики**

Как любой журнал или проспект теряет привлекательность без цветных иллюстраций, так и HTML-документ кажется невзрачным без использования графики. Под графикой подразумевают значки, рисунки, фотографии и карты-изображения, занимающие часть окна браузера. Графика привлекает внимание посетителя, заостряет его на наиболее важных моментах.

Для вставки графических изображений в HTML-документ используется специальный тег <img>, который не требует наличия закрывающего тега.

На сегодняшний день для создания графических изображений, предназначенных для размещения в электронных документах, используются три основных формата: GIF, JPG и PNG.

Тег <img> имеет ряд атрибутов, которые задают оформление добавленного на веб-страницу изображения:

src – обязательный атрибут указывающий путь (относительный или абсолютный) к файлу изображения.

Если путь к файлу задан простым указанием имени файла:

<img src=“logo.gif”>,

то при подобном задании значения src браузер станет искать файл logo.gif в том же каталоге, что и HTML-документ, в котором данный рисунок будет расположен. Обычно для графических изображений выделяется специальная папка – images, и тогда путь к файлу прописывается так:

<img src=“images/logo.gif”>;

border – задает толщину рамки, отображаемой вокруг рисунка.

Толщина указывается в пикселях, по умолчанию (если параметр пропущен) рамка не отображается. Если графическое изображение является гиперссылкой, то браузер автоматически отобразит вокруг него рамку определенного цвета (обычно синего) толщиной в 1 пиксел. Если необходимости в рамке вокруг графического указателя ссылки нет, следует установить значение параметра border, равное нулю;

width, height – определяют ширину и высоту встраиваемого графического изображения.

Значения ширины и высоты выводимого рисунка, как правило, задаются   
в пикселях. Если данные параметры не указаны, браузером используются оригинальные размеры рисунка. Если указано значение только одного из параметров, браузер самостоятельно изменит второй параметр, сохраняя пропорцию. Если указаны значения обоих параметров, браузер отобразит рисунок заданных размеров, даже если изображение при этом сильно исказится;

align – определяет выравнивание рисунка относительно текста и прочих элементов документа. Возможны следующие значения атрибута:

left – текст обтекает рисунок по правому краю (изображение слева);

right – текст обтекает рисунок по левому краю (изображение справа);

top – верхний край рисунка выравнивается по самому высокому элементу строки;

textop – верхний край рисунка выравнивается по самому высокому текстовому элементу строки;

absmiddle – выравнивание середины рисунка по середине строки;

bottom – выравнивание нижнего края рисунка по базовой линии строки;

hspace, vspace – определяют размер горизонтального и вертикального отступов от встраиваемого изображения до текста. Значения атрибутов задаются в пикселах. По умолчанию значения равны нулю;

alt – необязательный параметр, задает альтернативный (поясняющий) текст. Обычно это короткое описание изображения, которое появляется на мес­те рисунка в то время, пока идет загрузка изображения или если у пользователя отключена функция демонстрации графики;

title – отображает текст подсказки к рисункам при наведении курсора.   
Работает только при загруженном изображении.

Конструкция тега вставки рисунка в HTML-документ может иметь вид

<img src=“images/img1.jpg” align=“left” border=”1” width=“100” height=“150” hspace=“15” vspace=“15”>.

В соответствии с указанными параметрами, на странице появится изображение из файла img1.jpg, находящегося в папке images. Рисунок будет занимать площадку 100×150 пикселей у левого края окна браузера и обтекаться текстом с отступами по 15 пикселей сверху, снизу и справа. Надо заметить, что   
в данном случае, заданный параметром hspace горизонтальный отступ отодвинет рисунок на 15 пикселей от левого края документа.

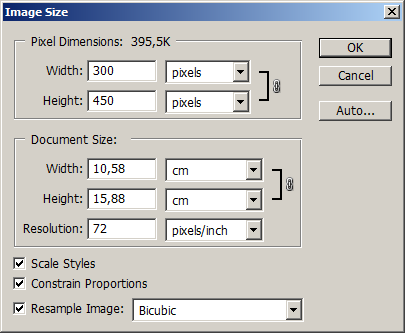
При добавлении графики на веб-страницу следует учитывать несколько факторов:

* размер файла изображения;
* разрешение рисунка (размер растровой сетки изображения);
* формат файла изображения;
* цветовую модель, в которой сохранено изображение.

О некоторых из них было сказано выше, остальные рассмотрим подробнее.

*Размер файла изображения*

Передача большого файла картинки при низкой пропускной способности канала ощутимо уменьшает скорость загрузки страницы и, кроме того, неоправданно увеличивает трафик. Так, например, если оригинальное изображение из файла img1.jpg имеет разрешение 300×450 пикселей, при этом размер файла составляет 7,24 кБ, то то же изображение с той же компрессией в требуемых габаритах 100×150 пикселей может иметь размер файла 1,87 кБ, то есть почти в 3,8 раза меньше.



*Рис. 42*. Диалоговое окно редактора Adobe Photoshop,   
предназначенное для изменения размеров изображения

Если на странице использовано несколько изображений, то размер файла каждого из них будет включен в суммарный размер загружаемой страницы, что, безусловно, отразится на трафике[[6]](#footnote-6). В этой связи уместно заранее уменьшать размер изображения в файле до значений, необходимых для верстки.

Для уменьшения размеров картинки можно воспользоваться растровым редактором Adobe Photoshop (рис. 42). В диалоговом окне Image Size в полях Width и Нeight нужно ввести требуемые значения ширины и высоты в пикселях. Изменяя размеры изображения, важно учитывать сохранение первоначальных пропорций изображения.

*Указание габаритов рисунка*

Если размеры изображения предварительно изменены до требуемых,   
то параметры width и height можно не указывать. Однако использование этих параметров все же рекомендуется, поскольку при указании в теге <img> значений ширины и высоты во время загрузки изображения браузер сразу зарезервирует на странице столько места, сколько необходимо для отображения рисунка. В противном случае, при загрузке под рисунок резервируется слишком мало места, а потом, когда браузер подставляет под маленькое пространство реальные размеры картинки, страница начинает «скакать».

*Сохранение пропорций*

Размещая на странице уменьшенный вариант изображения, используя указание требуемых габаритов атрибутами width и height, следует проследить за сохранением пропорций. Так, например, если размер встраиваемого изображения составляет 300×450 пикселей, а в конструкции тега указано width=“100” height=“180”, то отображенная браузером картинка будет заметно искажена – растянута по вертикали.

Чтобы уменьшенный рисунок отображался корректно и случайные искажения были предупреждены, можно указать в теге <img> только один из параметров (width и height), значение другого браузер изменит самостоятельно, сохраняя пропорции.

*Задание относительных размеров изображения*

В случаях, когда браузеру следует изменять размеры рисунка в соответствии с размерами открытого окна, можно задавать значения параметров width или height в процентах. Такой подход не рекомендован, поскольку некоторые браузеры не поддерживают такое обозначение размеров либо выводят их некорректно.

*Форматы файлов изображений*

На сегодняшний день для создания графических изображений, предназначенных для размещения на веб-страницах, используются три основных формата: jpg, gif, png, (wbmp, ico).

**JPEG (Joint Photographic Experts Group)**

Формат JPEG (*для публикации в сети используют расширение: .jpg*) был разработан для передачи фотографий между различными платформами. Благодаря возможностям хранить изображения с 24-битовой глубиной цвета и обеспечивать высокую степень компрессии данных, формат JPEG получил широкое распространение для отображения в сети иллюстраций фотореалистичного качества. Применяемый в JPEG алгоритм сжатия (собственно, JPEG – это как раз название метода сжатия) позволяет варьировать степень компрессии файла   
в зависимости от требуемого качества отображения иллюстрации.

Кроме того, формат JPEG поддерживает опцию *прогрессивный JPEG*, которая позволяет показывать картинку за несколько проходов до того, как она полностью загрузится на клиентский компьютер. В файл JPEG записываются несколько иллюстраций полного размера, но различного уровня качества. Сначала отображается рисунок низкого качества (и, соответственно, малого объема), а затем за несколько итераций (обычно догружаются 3–5 все более качественных изображений) он превращается в обычный (рис. 43). Это актуально при использовании файлов большого размера и низкой скорости связи, поскольку позволяет увидеть содержание изображения еще до его полной загрузки. При этом общий объем загружаемого файла возрастает несущественно   
по сравнению с оригинальным JPEG.

*а б*

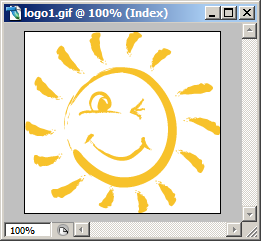
*Рис. 43.* Изображение фотографии на начальном этапе (*а*)   
и после окончания загрузки файла (*б*)

**GIF (Graphics Interchange Format)**

Формат GIF (*расширение: .gif*) был разработан компанией Compuserve Inc для передачи графической информации с использованием компьютерных сетей. В основе GIF лежит подход, основанный на том, что не каждое цветное изображение должно являться полноцветным. Это значит, что если количество цветов, участвующих в формировании растрового изображения, без существенной потери качества, может быть ограничено до 256, то можно изменить способ кодирования цвета такого изображения и тем самым существенно уменьшить размер графического файла. Таким образом, GIF поддерживает до 256 произвольных цветов, что делает его не слишком пригодным для передачи фотографических изображений, но является вполне достаточным для создания графических меню, фоновых текстур, различных рисованных элементов оформления веб-страницы.

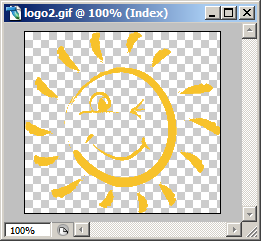
Формат GIF обладает рядом свойств, определивших его широкую популярность в веб-дизайне:

*Прозрачность.* Компьютерные изображения по своей природе всегда прямоугольны, но иногда требуется скрыть это, и тогда фон изображения делают совпадающим с фоном веб-страницы (рис. 44). Изображения в формате GIF могут иметь прозрачные области, что позволяет отойти от стандартных прямоугольных форм и создавать элементы страницы произвольной формы, которые могут размещаться на многоцветном фоне (рис. 45).

*а б*

*Рис. 44.* Белый цвет фона рисунка (*а*) совпадает с цветом фона страницы,   
поэтому прямоугольные границы картинки не виды (*б*)



*а*



*б*

*Рис. 45.* Прозрачный фон (*а*) скрывает прямоугольную форму рисунка,   
поэтому истинные границы картинки не читаются на цветном фоне веб-страницы (*б*)

*Чередование строк*. Графический файл может быть записан таким образом, что при отображении сначала будет загружаться каждая четвертая строка, растянутая на три соседних (изображение низкого качества). Затем, по мере загрузки последующих строк, изображение примет окончательный вид. Такая схема позволяет пользователю получить представление об изображении еще   
до окончательной загрузки графического файла.

*Анимация*. Форматом GIF предусмотрена возможность создания анимированных изображений. Для этого несколько слоев изображения, записанных   
в один файл, чередуются в определенной последовательности с заданным интервалом времени.

*Компактность.* Возможность ограничения цветовой палитры только цветами, участвующими в изображении, или преднамеренное уменьшение количества цветов до 2, 4, 8, 16, 32 и т. д. уменьшает объем сохраняемых данных. Наличие специального алгоритма сжатия данных позволяет подвергать компрессии файлы без изменения качества.

**PNG (Portable Network Graphics)**

Формат PNG (*расширение : .png*) разрабатывался для передачи графики по сети, с учетом всех особенностей Интернета. Он вобрал в себя преимущества предыдущих стандартов и исключил некоторые их недостатки.

Формат PNG поддерживает изображения как с полноцветной палитрой (PNG-24), так и с пользовательской палитрой цветов (PNG-8).

PNG поддерживает многоуровневую прозрачность, что позволяет создавать плавный переход от прозрачной области изображения к цветной (градиент).

Реализованный в PNG алгоритм сжатия позволяет сжимать рисованные изображения без потери качества лучше, чем GIF (от 5 до 25 %).

В PNG применяется механизм чересстрочного вывода первой картинки   
с использованием двумерной чересстрочности, что позволяет обеспечить минимальные затраты времени на вывод первого изображения.

Кроме того, в PNG реализован эффективный механизм контроля целостности файла, позволяющего обнаружить ошибки при передаче по сети.

Форматом PNG не поддерживается анимация.

Несмотря на некоторые преимущества данного формата, он не получил такого широкого распространения в Интернет, как форматы GIF и JPEG. Однако этот формат активно используется в специфических направлениях интернет-отрасли (например, в работе программного модуля GD::Graph, автоматически генирирующего графики и диаграммы для систем статистики, рейтинга и пр.).

Дополнительную информацию о добавлении изображений на веб-стра­ни­цу можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

Практическая работа

Добавление графики на веб-страницу. Особенности веб-графики

1. Откройте созданный ранее документ – Глоссарий терминов. Добавьте   
к текстовому содержимому веб-страницы иллюстрации по смыслу.

2. Поэкспериментируйте с размерами и размещением иллюстраций, варьи­руя значения атрибутов тега <img> и место вставки тега в HTML-документ.

3. Придайте веб-странице законченный вид, гармонично совместив иллюстрации и текст.

4. Добавьте к внутренним ссылкам, обеспечивающим быстрое перемещение вверх страницы, графические элементы – стрелки.

Форма отчета

Результатом выполнения работы является папка с файлами, включающая: основной HTML-документ и все связанные с ним файлы. Содержимое основного HTML-документа демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.

**Контрольные вопросы**

1. Опишите способ добавления графики на веб-страницу тегом img. Расскажите об особенности совместного использования атрибутов width и height тега img.
2. Какими характеристиками должен обладать графический документ, размещаемый на веб-странице?
3. Назовите форматы файлов графических данных, пригодные для размещения на веб-странице, область их применения.

## 3.5. Метаданные веб-документов

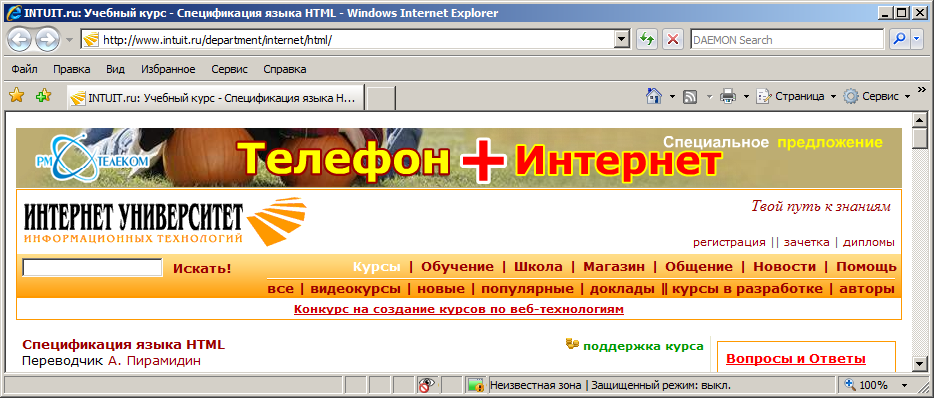
Внутри раздела head можно размещать теги HTML: <title>, <meta>, <link>.

Тег TITLE

Парный тег <title> … </title> используется для указания имени созданного документа. Он отображается как заголовок страницы или название окна документа.

<title>intuit.ru: Учебный курс - Спецификация языка HTML</title>

Под именем документа в данном случае имеется в виду не имя файла,   
а визуальный заголовок HTML-страницы (рис. 46).



*Рис. 46.* Содержимого тега <title> отображено в заголовке окна браузера   
и названии текущей вкладки

Тег <title> в HTML-документе можно использовать только один раз. Указание в заголовке конструкции <title> … </title> не является обязательным, однако рекомендуется по ряду причин:

* при отсутствии тега названия документа браузер при интерпретации HTML-кода выведет в заголовке окна фразу типа «Untitled Document» («Документ без названия»);
* при попытке добавить созданную без тегов <title> … </title> страницу   
  в «закладки» браузера пользователю придется самостоятельно вписывать название добавляемой страницы;
* поисковые системы, индексируя безымянный документ, занесут его   
  в базу под заголовком Untitled, что сделает документ безликим и затруднит его поиск через поисковые сервисы.

**Метаданные**

Метаданными называется информация о внутренних свойствах документа, а не о его содержимом. Спецификация метаданных состоит из двух шагов.

1. Объявление свойства и значения этого свойства;
2. Ссылка на профиль, где определены свойства и их действительные значения.

Объявить свойства и их значения можно двояко:

* задать их в текущем документе через элемент meta;
* объявить их во внешнем документе и связать метаданные через элемент link.

**Метаопределения META**

Метаданные могут вводиться в HTML-документ при помощи непарного тега <meta>. Все метаопределения имеют два основных типа данных: name   
и http-equiv. Атрибуты http-equiv и name являются двумя возможными параметрами тега <meta>, но поскольку они имеют разное предназначение, указывать их одновременно в рамках одного тега нельзя. <meta http-equiv …> переопределяет переданные сервером http-заголовки, <meta name …> устанавливает метаданные документа.

Структура метаданных типа – name. Конструкция метатега выглядит так:

<meta name=”имя” content=”содержание”>

Конструкции этого типа могут задавать:

* набор ключевых слов, предназначенных для индексирования документа поисковыми системами. Слова указываются через запятую:

<meta name=”keywords” content=”учебный курс, спецификация языка, html, intuit”>

* краткое описание данной веб-страницы, которое будет отображаться поисковыми системами на странице выдачи результатов рядом со ссылкой   
  на найденный интернет-ресурс:

<meta name=”description” content=”Обучающий курс построен по спецификации, которая определяет HTML 4.01 HyperText Markup Language (HTML) - гипертекстовый язык разметки, язык World Wide Web”>

* указание автора (авторов) текущего документа:

<meta name=”Author” content=”Иванов, Петров, Сидоров”>

* описание правил индексирования документа поисковыми роботами:

<meta name=”robots” content=” значение”>

Возможные значения параметра CONTENT: All, None, Index, Noindex, и др.

Если конструкция meta name=”Robots” не указывается, то поисковый робот индексирует документ со всеми присутствующими в нём гиперссылками, считая принятым по умолчанию значение All или Index, Follow.

Теперь рассмотрим метаопределения http-equiv. Конструкция метаданных выглядит аналогично предыдущей:

<meta http-equiv =”имя” content=”содержание”>

Значениями атрибутов HTTP-EQUIV можно задавать:

Expires – запрет кеширования HTML-страницы по истечении указанного срока;

Refresh – перезагрузка/переадресация через заданный промежуток времени;

Content-Type – определение типа и кодировки документа;

Content-Language – указание языка документа;

Canche-Control – контроль кеширования документа;

Window-target – определение места загрузки документа (для фреймовых структур);

Pragma – управление кешированием по протоколу HTTP/1.0;

Set-Cookie – настройка чтения данных Cookies пользовательского компью­тера;

Ext-cache – управление альтернативным кешем документа;

Location – указание полного адреса расположения документа в сети   
Интернет.

Рассмотрим применение некоторых конструкций.

<meta http-equiv =” expires” content=”Mon, 28 Dec 12:00:00 GMT”>

Данная конструкция указывает дату устаревания документа и запрещает кеширование страницы после указанного времени. При повторном обращении   
к документу по истечении заданного в параметре content срока браузер будет перенаправлен к источнику для обновления информации. Значение параметра content содержит следующую последовательность: день недели (Mon, Tue   
и т. д.), число (01, 02, 03 и т. д.), месяц (Jan, Feb, и т. д.), год, время (чч:мм:сс)   
и часовой пояс (GMT).

<meta http-equiv =”refresh” content=”10; URL=http://www.omgtu.ru”>

Такая конструкция предписывает браузеру перезагружать содержимое окна через заданный промежуток времени (в секундах). Если дополнительно добавлен подпараметр URL, то через указанное время будет произведена переадресация на указанный внешний адрес.

<meta http-equiv =” content-type” content=”text/html; charset=Windows-1251”>

Эта конструкция определяет тип и кодировку документа. Наиболее используемыми значениями кодировки русскоязычных документов являются Windows-1251 и KOI8-R. Кодировка указывается посредством подпараметра charset. На некоторых серверах возможно автоматическое перекодирование документа, в связи с чем использовать эту конструкцию не рекомендуется. Важно, чтобы кодировка текста документа совпадала с кодировкой, обозначенной   
в значении charset.

Дополнительную информацию об описании элемента META можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

**Связь между документами LINK**

Если веб-страница имеет структурные или функциональные связи с другими документами, то для описания этих связей используется непарный тег <link>. В конструкции этого тега могут использоваться следующие атрибуты:

href – указание ссылки на внешний документ;

type – тип внешнего документа;

rel – отношение между текущим и внешним документом;

rev – отношение между внешним и текущим документом.

Указать на связь текущей веб-страницы с внешним файлом шаблона стилей main.css можно, поместив в разделе head конструкцию:

<link href=”styles/main.css” type=”text/css” rel=”stylesheet”>.

Дополнительную информацию об описании элемента LINK можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

Практическая работа

Метаданные веб-документов

1. Для выполнения работы необходимо выбрать веб-страницу рекламно-информационного сайта, размещенного в сети Интернет. Для этого, используя любую поисковую систему, введите ключевые слова запроса и осуществите поиск ресурса по выбранной теме. Зафиксируйте для предстоящего анализа следующие сведения: введенные в строку поиска ключевые слова; заглавие и текст описания страницы, которая показалась наиболее соответствующей поисковому интересу. Сделайте и сохраните скриншот страницы поисковой системы, содержащей результат выдачи.

2. Перейдите по ссылке на выбранную веб-страницу. Сравните текст заголовка открытой веб-страницы с текстом заголовка, размещенного в листе выдачи поисковой системы, и с текстом введенного поискового запроса. Сделайте и сохраните скриншот веб-страницы.

3. Откройте исходный код выбранной веб-страницы. Сравните текст заголовка с содержимым тега-контейнера <title>. Сравните текст описания страницы в результатах выдачи с содержимым META-тега, включающего атрибут name=”description”, а также с текстом введенного поискового запроса. По результатам сравнения сделайте вывод.

4. Рассмотрите содержимое META-тега, включающего атрибут name=”keywords” в исходном коде страницы. Обратите внимание, совпадают ли введенные в поисковый запрос слова с заявленными в значении атрибута content. По результатам сравнения сделайте вывод.

5. Рассмотрите содержимое остальных МЕТА-тегов данной веб-стра­ни­цы. Расшифруйте все метаопределения, заданные для этой страницы. Опишите полученные результаты в тексте отчета. Скопируйте фрагмент исходного кода страницы, включающий описание раздела заголовка (<head>).

6. Создайте документ с описанием хода и результатами выполнения данной практической работы. Сформулируйте и запишите вывод.

Форма отчета

Результат выполнения работы представляется в форме распечатанного текстового документа, содержащего следующие элементы:

* скриншот страницы выдачи поисковой системы, содержащий поле   
  с текстом введенного запроса и блок с описанием страницы, выбранной для анализа;
* скриншот открытой страницы, выбранной для анализа;
* листинг фрагмента HTML-кода страницы (раздел заголовка), выбранной для анализа.

Контрольные вопросы

1. Что называют метаданными документа? Назовите теги описания метаданных HTML-документа.
2. Какая информация об HTML-документе описывается тегом META? Опишите конструкцию тега МЕТА.
3. Какую информацию, содержащуюся в HTML-коде страницы, используют поисковые системы при обработке поискового запроса?

## 3.9. Стилевое оформление веб-страницы с использованием CSS

Спецификация языка разметки HTML позволяет разработчику электронных документов изменять внешний вид некоторых элементов страниц. Для этого составляются специальные правила отображения конкретного элемента, называемые каскадными таблицами стилей (CSS Cascading Style Sheets) или стилевыми шаблонами:

*каскадная* – спецификация HTML разрешает использовать для одного   
и того же элемента несколько стилевых правил, интерпретируемых браузером последовательно, то есть каскадом;

*таблица* – формат записи стилевых правил напоминает табличное представление данных: заголовок таблицы соответствует наименованию элемента, класса или идентификатора стиля; в качестве ячеек и рядов выступают стилевые свойства и их значения;

*стилей* – под стилем понимают приведение какого-либо явления к общему набору правил и определений. В этом смысле CSS – способ дополнительного форматирования стандартных тегов HTML.

Пример записи стиля оформления заголовка 3-го уровня:

h3 { color: blue; font-family: Tahoma, Verdana, Arial; }

или в другом виде:

h3

{

color: blue;

font-family: Tahoma, Verdana, Arial;

}

**Способы определения таблиц стилей**

Любая таблица CSS должна быть интерпретирована браузером для того, чтобы правила CSS, обозначенные для конкретных элементов, вступили в силу.

Определение таблиц стилей возможно четырьмя способами:

*1. Ссылка на внешний файл*

Если все стилевые шаблоны разместить в одном текстовом файле (с расширением .css), то с помощью тега <LINK> из текущего докумена можно сделать ссылку на внешний CSS-файл стилевых шаблонов, например:

<link rel=”stylesheet” type=”text/css” href=”style.css”>.

Конструкция указания пути к файлу находится в пределах раздела HEAD. Браузер, анализируя HTML-код, обратится по указанному пути и, обнаружив указанный файл, отобразит элементы в соответствии с указанным в файле   
стилем.

В случае ошибки интерпретации HTML-кода браузером, плохой связи   
с сервером и пр., внешний файл CSS может не загрузиться, вследствие чего стиль для нужных элементов HTML не будет переопределен.

Если внешний файл включает слишком большое количество стилевых шаблонов (что скажется на размере файла), то существует вероятность того, что браузер не сумеет полностью интерпретировать файл CSS или вообще исчер­пает лимит времени по загрузке данных. В первом случае стили части элементов не будут переопределены (браузер успеет обнаружить только правила, размещенные в верхней части CSS-файла); во втором случае все элементы страницы останутся без изменения, загрузившись по умолчанию.

*2. Задание стилевой конструкции внутри HTML-страницы*

< style type=”text/css”>

<--!

body { font-family: Tahoma, Verdana, Arial;}

input { background-color: #cecece;}

-->

</style>

Данная конструкция также должна присутствовать в разделе HEAD. Описание стиля заключается между символами комментариев с тем, чтобы браузеры, не поддерживающие CSS , пропускали содержание стилевых шаблонов. Браузеры, поддерживающие CSS интерпретируют код и отобразят страницу   
в соответствии с указаниями.

Наличие слишком большого количества шаблонов CSS заметно увеличивает размер страницы, что скажется на времени загрузки документа в браузере.

*3. Включение в теговую конструкцию отдельного HTML-элемента*

<p align= “justify” style=“color: #00ffcc; font-family: Tahoma, Verdana, Arial;”>

…текст параграфа...

</p>

*4. Импортирование стилевого шаблона (ссылка на внешний файл в Интернет)*

< style type=”text/css”>

<--!

@import: url (style.css);

-->

</style>

При импортировании стилевого шаблона оператор импорта стиля должен предшествовать всем прочим описателям стилей:

<style>  
@import: url(http://kuku.ru/style.css)  
а {color:cyan;text-decoration:underline;}  
</style>

Все четыре способа определения стилевого шаблона можно использовать одновременно в пределах одного HTML-документа.

При сочетании различных способов следует учитывать особенности браузеров.

**Правила составления стилевых шаблонов**

Любое правило оформления стиля состоит из *селектора* и *определения шаблона*.

*Селектор* – это условие отображения стиля для заданного элемента или наименование внутреннего класса (идентификатора) для него.

*Определение шаблона* – это описание стилевых правил оформления, которые чередуются через точку с запятой и заключаются в фигурные скобки.

H3 { color: blue; font-family: Tahoma, Verdana, Arial; }

Определение (определят правило отображения цвета заголовка и правило отображения гарнитуры)

Селектор

Возможно наследование стилевых правил для нескольких селекторов одновременно, например:

td, th, p, h3, div {color: blue; font-family: Tahoma, Verdana, Arial; font-size: 10px;}

В качестве селектора CSS могут выступать:

* *элементы HTML* (будет определяться стиль для конкретного элемента страницы):

body { color: blue; }

Весь текст в пределах BODY будет голубым, но при добавлении, например, таблицы, назначение стилевого шаблона пропадает для текста внутри   
ячеек;

* *классы* (использование классов позволяет переопределять стиль для любого элемента, которому присвоен данный класс).

Наименование класса начинается с точки и пишется строчными буквами, например:

.reg { color: blue; }

В этом случае все элементы, которым присвоен класс.reg,будут менять цвет на голубой:

<font class=”reg”> .... </font>

Если дополнить селектор класса наименованием конкретного элемента, то действие стилевого правила будет распространяться только на этот элемент:

hr.reg { color: blue; }

* *идентификаторы* (использование идентификаторов позволяет переоп­ределять стиль для любого элемента так же, как использование классов. Однако селектор идентификатора часто применяется для задания уникального имени элементу, который задействован в программном сценарии (скрипте)).

Запись идентификатора начинается с символа # и заканчивается наименованием.

#reg { color: blue; }

В этом случае элемент, которому присвоен идентификаторreg,будет менять цвет на голубой.

<td id=”reg”> .... </td >

При задании стиля документа возможно использование псевдоклассов. *Псевдоклассами* называют определенные условия форматирования, в соответст­вии с которыми браузер подставляет необходимые стилевые правила отображения данных. При этом в исходной структуре документа такие классы не присутствуют: они создаются в процессе интерпретации HTML-кода браузером.

Псевдоклассы могут использоваться для задания различных типов форматирования по нескольким разновидностям элементов.

Например, если рассматривать гиперссылку, то её отображение в зависимости от состояния (активная, посещённая, непосещённая, наведён курсор) может быть различным. Цвета отображения состояния ссылки можно прописать   
в теге BODY, а можно задать псевдоклассами при записи в стилевом шаблоне:

a: link { color: blue; } – отображение непосещенной ссылки;

a: active { color: red; text-decoration: underline; } – отображение активной ссылки;

a: visited { color: gray; } – отображение посещенной ссылки;

a: hover { color: green; } – вид ссылки при наведении курсора мыши.

Если есть необходимость визуально выделить одни ссылки относительно других, наряду с псевдоклассами используют обычные селекторы классов. Например:

a: link.red { color: blue; };

a: active.red { color: red; text-decoration: underline; };

a: visited.white { color: gray; };

a: hover. white { color: green; }.

**Наследование и переопределение стилей**

1. Сначала применяются стили умолчания браузера.
2. Стили умолчания браузера переопределяются прилинкованными стилями (элемент LINK заголовка документа).
3. Прилинкованные стили переопределяются описаниями стилей в элементе STYLE.
4. Стили элемента STYLE переопределяются атрибутом STYLE в любом из элементов разметки.

1. Приступая к работе, продумайте, как будут совмещаться левая и правая, а также верхняя и нижняя стороны создаваемого рисунка. [↑](#footnote-ref-1)
2. Скриншот (screenshot) – снимок экрана, изображение, полученное компьютером, которое точно отображает то, что показывает монитор в конкретный момент времени. [↑](#footnote-ref-2)
3. В раздел <body> демонстрируемых страниц разумно добавить несколько абзацев произвольного текста, чтобы сделать вывод о пригодности фона, читабельности текста на данном фоне. [↑](#footnote-ref-3)
4. Акронимы – некие устоявшиеся сокращения, например СНГ, США. [↑](#footnote-ref-4)
5. Глоссарий (лат. glossarium – «собрание глосс») – словарь узкоспециализированных терминов в какой-либо отрасли знаний с толкованием, иногда переводом на другой язык, комментариями и примерами. [↑](#footnote-ref-5)
6. Трафик – количество данных, которые проходят через сервер за определенное время. Измеряться он может в байтах, килобайтах, мегабайтах и т.д. [↑](#footnote-ref-6)