МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

ИНСТИТУТ ХОЛОДА И БИОТЕХНОЛОГИЙ

Е.Б. Петрунина, Е.Г. Селина

ОСНОВЫ HTML

Учебно-методическое пособие

Санкт-Петербург
2013

УДК 681.3

Петрунина Е.Б., Селина Е.Г. Основы HTML: Учеб.-метод. пособие. – СПб.: НИУ ИТМО; ИХиБТ, 2013. – 49 с.

Подробно рассмотрены возможности основных тегов языка HTML. Даны сводные таблицы тегов и их атрибутов. Приведены задания для самостоятельного закрепления изложенного материала.

Учебно-методическое пособие предназначено для студентов всех направлений (бакалавриат) очной формы обучения.

Рецензент: кандидат техн. наук, проф. А.В. Зайцев

Рекомендовано к печати редакционно-издательским советом Института холода и биотехнологий

В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики».

ВВЕДЕНИЕ

Создание программных продуктов, имеющих удобный интерфейс для разработки документов коллективного пользования, а также для работы с документами, уже опубликованными в сети, снизило уровень начальной подготовки, необходимой для доступа к сетевой информации. Тем не менее, полезно знать основные принципы составления таких документов и связанную с ними терминологию.

В первом разделе методических указаний подробно изложены основные положения стандарта HTML, который используется для создания WEB-страниц, правила составления тегов для форматирования шрифта, компоновки материала на WEB-страницах и объединения их в сайт. Приведено много примеров, иллюстрирующих использование тегов в разных ситуациях.

Во втором разделе приведены задания для самостоятельного закрепления изложенного материала.

В Приложении приведены краткие сведения по основным тегам и атрибутам. Они могут быть полезны как справочник при самостоятельном составлении HTML-страницы или корректировке документов, созданных стандартными редакторами.

Более полное знакомство с возможностями оформления сетевых документов можно получить из источников, представленных в списке литературы [1–5].

1. ОСНОВНЫЕ ТЕГИ HTML

1.1. Понятие об HTML

Гипертекстовые документы создаются на специальном языке, который называется HTML (Hyper Text Markup Language). Разработка этого стандарта позволила снять две существенные проблемы, которые возникали у пользователей больших сетей.

Первая из них – время доставки информации. Оно существенно зависит от объема передаваемого файла. Как правило, пользователь работает в каждый момент с небольшим объемом информации. При стандартной организации файла «все в одном» приходится ждать, пока загрузятся не только нужные, но и все сопутствующие сведения, а затем «копаться» в большом файле в поисках того, что нужно. В стандарте HTML вся информация по некоторой теме разбивается

на небольшие смысловые блоки и посылается пользователю только по мере возникновения необходимости в ней. Это существенно сокращает время ожидания ответа на возникший вопрос и делает работу в сети более комфортной и дешевой. Для организации такого режима работы в языке HTML введено средство *гиперссылка* (см. п. 1.12).

Вторая проблема – распознавание пришедшей по запросу информации. Способы кодировки одних и тех же сведений в разных текстовых редакторах часто бывают разными. И если компьютер-получатель оснащен не тем текстовым редактором, на котором создавался документ, он не всегда может интерпретировать пришедшую на него информацию. Чтобы преодолеть это, в языке HTML установлены следующие правила:

– весь документ набирается только текстовыми символами (их коды понятны всем редакторам), т. е. его можно набрать в любом текстовом редакторе (например, "Блокнот", "WordPad");

– символы и конструкции, которые непонятны браузеру, игнорируются (а текстовые редакторы обычно прекращают интерпретировать документ, если встречают в нем непонятные коды);

– в текст документа можно добавлять разнообразные команды, которые указывают, как следует изображать на экране тот или иной фрагмент текста, или обеспечивают спецэффекты. Эти команды называют *тегами.* Они так же, как и документ, набираются текстовыми символами, но отделяются от него угловыми скобками: < >;

– файлы с HTML-кодом должны помечаться расширением .html или .htm

1.2. Структура тегов

Каждый тег обязательно содержит пару угловых скобок < >, между которыми находится *ключевое слово*. Оно обозначает основную функцию тега и размещается вплотную к открывающей скобке. Например, ключевое слово HR в теге <HR> означает, что тег вставляет в документ горизонтальную линию (Horizontal Row)

Действие тега детализируется с помощью *атрибутов*. Например, применительно к тегу <HR> можно уточнить, какими должны быть длина, толщина и цвет линии. Каждому ключевому слову соответствует свой собственный набор атрибутов и стандартные значения для них. Например, для тега <HR> стандартными являются серый

цвет, длина по всей ширине страницы и толщина в один пиксель. Эти значения автоматически присваиваются тем атрибутам, которые не внесены в тег, поэтому в него можно включать только те атрибуты, значения которых Вы хотите изменить.

Атрибуты размещают после ключевого слова и перечисляют в произвольном порядке через пробел. После каждого атрибута ставят знак = и нужное значение. Например, тег <HR WIDTH=50% SIZE=4> означает, что в документ следует вставить горизонтальную линию, которая должна доходить только до середины строки и иметь ширину не один, а четыре пикселя. Значения атрибутов полагается заключать в двойные кавычки. Однако в современных браузерах это требование обязательно только для тех значений, которые не укладываются в одно слово (например, название файла в гиперссылке).

Атрибуты, соответствующие одним и тем же свойствам, в разных тегах часто, но не всегда, обозначаются одинаковыми словами. Например, выравнивание текста в пределах строки, ячейки таблицы или около картинки задается одинаковым словом ALIGN. А свойство «цвет шрифта» для документа в целом задается атрибутом TEXT, но для отдельно взятого фрагмента документа – атрибутом COLOR. Соответствующие теги, задающие темно-синий цвет шрифта, выглядят так:

<BODY TEXT=NAVY>, <FONT COLOR=NAVY>.

Многие теги являются парными. Открывающий тег включает какое-нибудь действие, закрывающий – прекращает его. В этом случае ключевое слово у обоих тегов одинаковое, но в открывающем теге дополнительно перечисляются все необходимые атрибуты, а в закрывающем атрибутов нет, и перед ключевым словом ставится знак /. Парные теги называются *контейнерами*. Например, контейнер

<B>Я изучаю язык HTML</B>

означает, что текст Я изучаю язык HTML следует изобразить жирным шрифтом, а дальнейшие символы – так же, как до него.

Замечания:

– в ключевых словах и атрибутах строчные и прописные регистры букв не различаются, однако для облегчения редактирования рекомендуется набирать их заглавными буквами;

– URL файлов-ссылок заключаются в двойные кавычки, и регистры букв в них учитываются;

– в последовательности пробелов, идущих подряд, учитывается только первый, остальные игнорируются;

– признак начала новой строки игнорируется. Поэтому при составлении кода можно клавишей <Enter> разрывать его на строки так, чтобы код удобно было просматривать и редактировать. На виде документа в окне браузера это не отражается. Если нужно сохранить при просмотре через браузер введенные в код HTML пробелы и разрывы строк, следует воспользоваться тегами компоновки текста (см. п. 1.7).

Списки основных тегов, их функций и атрибутов приведены в прил., табл. 1–4. Пояснения по основным группам тегов даны в п.п. 1.5–1.14.

1.3. Кодирование цвета в HTML-стандарте

В некоторых тегах HTML-стандарта используются атрибуты, влияющие на цвет изображаемого объекта (теги <HR>, <FONT>, <BODY> и другие). Для того чтобы задать нужный цвет, можно использовать его название на английском языке или указать его код в системе RGB.

Метод (система) RGB (True color) – от слов Red, Green, Blue – удобен для изображений, пересылаемых по сети и рассматриваемых на экране или выводимых на устройство записи на киноплёнку.

Оттенки цвета создаются смешением лучей трёх базовых цветов разной интенсивности. Коды интенсивностей каждого луча записываются в шестнадцатеричной системе, которая использует в качестве цифр символы 0, 1, 2, 3, 4, …, 9, A, B, C, D, E, F.

Примеры записи чисел в шестнадцатеричной системе:

$5_{10} = 5_{16}$, $13_{10} = C_{16}$, $17_{10} = 11_{16}$, $47_{10} = 2F_{16}$, $255_{10} = FF_{16}$.

Под значение *интенсивности каждого луча* отводится 1 байт, т. е. можно использовать 256 разных уровней интенсивности. Для совокупности трёх лучей получается $256^3 = 16\,777\,216 \approx 17$ млн оттенков.

Для записи кода выбранного цвета указываются шестнадцатеричные коды интенсивности каждого базового луча в последователь-

ности, отраженной в аббревиатуре RGB. Под каждую составляющую кода отводят две позиции. Перед общим кодом размещается знак #.

Примеры:

#000000 – нет ни одного цвета (чёрный);

#FFFFFF – все цвета в максимальной интенсивности (белый);

#3F3F3F – все лучи одинаковой промежуточной интенсивности (один из оттенков серого);

#CF35D1 – номера интенсивностей каждого из лучей: красного – $CF_{16} = 12*16^1+15*16^0=207_{10}$, зелёного – $35_{16} = 3*16^1+5*16^0=53_{10}$, синего – $D1_{16} = 13*16^1+1*16^0=209_{10}$.

В десятичной системе коды интенсивностей лучей для выбранного цвета можно узнать в диалоговом окне Цвета на вкладке Спектр, которое вызывается командой Заливка. Для перевода этих кодов в шестнадцатеричную систему удобно воспользоваться программой Калькулятор в инженерном режиме.

1.4. &-последовательности, комментарии

Иногда в документе приходится использовать такие знаки, которые имеют в языке HTML специальный смысл. Например, двойные кавычки " " выделяют значение атрибута, и если их использовать вне тега, браузер не поймет конструкцию и пропустит ее. Чтобы этого не произошло, для отображения в документе символов, которые имеют служебное значение, используют специальные последовательности знаков, начинающиеся с символа & и заканчивающиеся знаком ; (точка с запятой) – так называемые *&-последовательности*. Основные из них:

– последовательность < (от англ. *less than*) – вместо символа < (меньше), так как этот символ браузер рассматривает как начало тега;

– последовательность > (от *greater than*) – вместо символа > (больше), так как этот символ браузер рассматривает как конец тега;

– последовательность " (от *quotations mark*) – вместо символа " (двойные кавычки);

– последовательность & (от *ampersand*) вместо символа &;

– последовательность   (от *nonbreaking space* – неразрывный пробел) – вместо пробела, который не следует игнорировать при выдаче документа в окно браузера. Эта &-последовательность как бы вставляет в текст невидимую глазу букву, которую браузер

воспринимает как реальную и оставляет для нее место на экране. Она используется для заполнения пустых ячеек таблицы, пустых абзацев, для склеивания слов (если набрать TOP 100, то у вас никогда не получится, что TOP останется в одной строке, а 100 перенесется на следующую, например, при изменении шрифта или размера окна браузера).

Внимание! &-последовательности должны набираться только в нижнем регистре (никаких < или " быть не должно!), и точка с запятой в конце последовательности обязательна. Пример: чтобы пользователь увидел на экране слова крейсер "Аврора", в HTML-документе должен быть следующий код:

крейсер "Аврора"

Чтобы код HTML было легче понимать и править, полезно вводить в него комментарии, которые были бы видны только при работе с кодом, но не отражались бы в окне браузера при просмотре документа. Для этой цели используют специальный тег, написанный не по стандартным правилам. Он выглядит так: <!--*пояснительный текст* -->

Примеры:

```
<!-- Это мой комментарий -->
<!-- А этот комментарий занимает
 в тексте несколько
строк -->
```

Внимание! В текст комментария не допускается вставлять два и более дефисов подряд: -- или ----- (разные браузеры по-разному реагируют на это).

1.5. Структура документа HTML

Каждый HTML-документ состоит из трех главных частей:

1) Объявление HTML-кода – контейнер <HTML> …</HTML>
2) Заголовок документа – контейнер <HEAD> … </HEAD>
3) Тело документа – контейнер <BODY> … </BODY>

Заголовок и тело документа вкладываются внутрь объявления HTML по следующей схеме:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
</BODY>
</HTML>
```

В заголовок помещаются теги, определяющие информацию о документе в целом. Наиболее употребительные из них – контейнер <TITLE> и тег <META>.

Общий вид контейнера <TITLE>:

<TITLE> Краткая расшифровка содержания документа или ключевые слова по нему (не более 64 символов) </TITLE>.

Пример:

<TITLE> Гостиницы для участников конференции </TITLE>

Без контейнера <TITLE> в строке заголовка браузера повторяется адрес активной страницы, который также выведен в поле адрес. Если в заголовке страницы есть контейнер <TITLE>, то текст, размещенный в нем, заменяет эту информацию. При просмотре большого количества файлов это облегчает ориентировку среди них.

Тег <META> – одиночный. Он содержит информацию о документе, которая нужна для браузера, поисковых систем Интернета и других приложений. В нем можно использовать следующие атрибуты:

NAME, HTTP-EQUIV – для указания типа информации, описание которой расшифровывается в атрибуте CONTENT этого же тега.

Примеры:

<META NAME=keywords CONTENT=”тег, гипертекст, браузер, HTML” > – этот тег задает ключевые слова по теме документа.

<META NAME=description CONTENT=”Описание возможностей языка HTML”> – этот тег задает краткое описание содержания документа.

<META HTTP-EQUIV=refresh CONTENT=60 URL=www.pp.ru/price.htm> – этот тег заставит браузер тратить на поиски гиперссылки с указанным URL не более 60 с.

<META HTTP-EQUIV=refresh CONTENT=10> – этот тег заставит браузер автоматически обновлять документ каждые 10 с.

<META NAME=author CONTENT="Петров Иван Сидорович"> – такой тег облегчает поиск документов данного автора.

В разделе «тело документа» размещается содержание документа, которое выдается в рабочее окно браузера. Атрибуты тега <BODY> задают следующие свойства:

TEXT – цвет текста там, где он не указан специальными средствами. По умолчанию черный.

BGCOLOR – цвет фона. По умолчанию белый.

BACKGROUND – фоновое изображение (аналогично рисунку на Рабочем столе). Значением является URL файла-изображения.

LINK,VLINK, ALINK – соответственно цвета не посещенных, посещенных и активных в данный момент гиперссылок.

Примеры:

```
<BODY TEXT=SILVER  BGCOLOR=NAVY>
<BODY TEXT=SILVER  BACKGROUND="море.jpg">
```

Первый тег задает для базового оформления документа светло-серый шрифт на темно-синем фоне. Все отступления от этого стандарта в дальнейшем надо будет оговаривать дополнительными тегами внутри документа. Второй в качестве фона использует изображение, находящееся в файле море.jpg, расположенном в том же каталоге, что и вызывающий его документ (подробнее см. п. 1.11). Цвет шрифта так же, как и в первом примере, – светло-серый.

Основные теги, определяющие вид документа в окне браузера, рассмотрены ниже и приведены в виде справочника в Приложении. С остальными можно ознакомиться, например, в [3–5].

1.6. Теги форматирования шрифта

В HTML существуют два подхода к шрифтовому оформлению текста – так называемые физические и логические стили. Здесь рассматриваются только физические стили. Под ними подразумевают прямое указание браузеру на изменение текущего шрифта. Теги фи-

зических стилей – контейнерные. Например, между тегами <B> и </B> будет жирный шрифт (Bold), а между <I> и </I> – курсив (наклонный – Italic).

Основные контейнеры физических стилей:

<B> … </B> – **жирный шрифт**;
<I> … </I> – *курсив*;
<U>… </U> – подчеркнутый текст;
<STRIKE> … </STRIKE>, <S> … </S> – ~~перечеркнутый шрифт;~~
<BIG> … </BIG> – шрифт чуть большего размера, чем текущий;
<SMALL> … </SMALL> – шрифт чуть меньшего размера, чем текущий;
_… – подстрочный $_{\text{индекс}}$;
[…] – надстрочный $^{\text{индекс}}$;
<FONT> … </FONT> – изменение типа, цвета и размера шрифта.

В теге <FONT> допускаются атрибуты:

SIZE – размер букв. Возможные значения: 1, 2, 3,…, 7. По умолчанию используется размер 3. Значение этого атрибута можно задавать также относительно стандартного. Тогда оно задается со знаком плюс или минус.

Примеры:

SIZE=+2 эквивалентно SIZE=5, SIZE=-2 эквивалентно SIZE=1;

COLOR – цвет шрифта. Можно указывать шестнадцатеричными кодами (см. п. 1.3) или английскими словами (см. табл. 3 Приложения). По умолчанию BLACK;

FACE – изменение типа шрифта, возможные значения: ARIAL, COURIER и т. д. По умолчанию VERDANA.

Теги могут комбинироваться, но при этом надо следить, чтобы контейнеры имели вложенную структуру (контейнер, открытый позже, должен быть закрыт до того, как будет закрыт внешний). Например, для комбинации жирного и наклонного шрифтов можно использовать такой код:

<B> <I> пример </I> </B>

В окне браузера мы получим: ***пример***.

Помимо перечисленных выше тегов существуют шесть контейнерных тегов, удобных для оформления заголовков: <H1>...</H1>, <H2>...</H2>, ... , <H6>…</H6>. Заголовки отличаются от текста жирностью букв и увеличением интервала по вертикали между заголовком и прочей информацией, а между собой – высотой букв (в заголовке <H1> самые крупные символы, в <H6> – самые мелкие). Атрибут ALIGN управляет выравниванием заголовков по горизонтали. По умолчанию он принимает значение LEFT. Другие возможные значения: CENTER, RIGHT.

Пример:

```
<H4 ALIGN=CENTER> 1.7. Теги выделения
 смысловых частей текста </H4>
```

Результат действия тега похож на заголовок следующего параграфа, но изображенный полужирным шрифтом.

1.7. Теги выделения смысловых частей текста

Абзацы в тексте выделяются контейнером <P>…</P> (Paragraph). Атрибут ALIGN задает выравнивание по горизонтали. По умолчанию он принимает значение LEFT. Другие возможные значения: CENTER, RIGHT. При просмотре браузером абзацы автоматически отделяются друг от друга пустой строкой. Закрывающий тег </P> необязателен. Его можно использовать только тогда, когда было заказано нестандартное выравнивание текста для одного или нескольких абзацев.

Примеры:

```
<P> Этот абзац расположен слева </P>
<P ALIGN=CENTER> Этот абзац по центру </P>
<P ALIGN=RIGHT> А вот этот абзац справа </P>
```

Приведенный выше код при просмотре выглядит так:

Этот абзац расположен слева

Этот абзац по центру

А вот этот абзац справа

Чтобы перейти на новую строку, не вставляя пустую строку в текст, применяется тег
 (Break). Закрывающего тега у него нет.

Иногда полезно разделять смысловые части документа горизонтальной чертой. Это делается с помощью одиночного тега <HR> (Horizontal Row). Его атрибуты:

SIZE – толщина линии в пикселях. По умолчанию 1 пиксель.

WIDTH – длина линии в пикселях или процентах от ширины экрана. По умолчанию 100 %.

ALIGN – выравнивание линии, возможные значения: LEFT, CENTER, RIGHT. По умолчанию LEFT. Этот атрибут имеет смысл заказывать в паре с WIDTH.

COLOR – цвет линии (возможные значения см. в прил., табл. 3). По умолчанию GRAY.

NOSHADE – линия рисуется без тени. По умолчанию тень есть.

Пример. Код, в котором использованы теги
 и <HR>:

```
<B> Жизненная позиция </B>
<HR  NOSHADE  WIDTH=250>
Хорошо на свете жить, <BR>
Информатику учить! <BR>
<HR  SIZE=4  WIDTH=50%  ALIGN=CENTER>
```

В окне браузера этот код изображается так:

Жизненная позиция

Хорошо на свете жить,
Информатику учить!

Контейнер <BLOCKQUOTE> … </BLOCKQUOTE> используют для того, чтобы добавить к выделенному им тексту поля со всех четырех сторон (сверху и снизу – по пустой строке, слева и справа – шириной в 5 знаков). Примером того, как выглядит текст, обработанный этими тегами, является следующий абзац.

> Еще один полезный контейнер – <PRE>…</PRE> (PREformatted text – предварительно форматированный текст). Обычно браузеры при выдаче текста на экран

учитывают в цепочке идущих подряд пробелов только первый и игнорируют символы конца строки. Однако то, что находится в контейнере <PRE>, выводится в окне браузера точно так, как оно расположено в коде HTML-документа.

Пример. Текст кода оформлен так:

```
<PRE>
                Чтобы стихотворение красиво
                    смотрелось в окне браузера,
                        вовсе не обязательно использовать
                            тег &lt;BR&gt;!
</PRE>
```

В окне браузера текст выглядит следующим образом:

```
Чтобы стихотворение красиво
    смотрелось в окне браузера,
        вовсе не обязательно использовать
            тег <BR>!
```

Так как выравнивание по центру используется чрезвычайно часто, введен специальный контейнер <CENTER>..</CENTER>. Он позволяет отцентрировать сразу несколько заголовков и абзацев, не используя дополнительно атрибут ALIGN.

1.8. Оформление списков

Списки облегчают нахождение нужной информации при перечислении сведений, объединенных общим смыслом: пункты меню, списки товаров, перечень сотрудников и т. п. В языке HTML предусмотрено несколько вариантов оформления списков, которые чаще всего используются в документации. Здесь будут рассмотрены только маркированные и нумерованные списки.

Для того чтобы оформить фрагмент документа в виде маркированного списка, его помещают в контейнер <UL>… </UL> (Unordered List). Для нумерованного – в контейнер <OL> … </OL> (Ordered List). Затем текст в контейнере разбивается на отдельные пункты списка тегами <LI> (List Item – элемент списка). Закрывающий тег </LI> в конце каждого пункта ставить необязательно. В мар-

кированном списке все пункты автоматически помечаются одинаковыми маркерами, в нумерованном – элементы нумерует сам браузер.

Примеры кода:

```
UL>Список коней:
    <LI>Сивка
    <LI>Бурка
    <LI>вещая Каурка
</UL>
```

```
<OL>Список коней:
    <LI>Сивка
    <LI>Бурка
    <LI>вещая Каурка
</OL>
```

Это дает на экране следующие списки:

Список коней:

- Сивка
- Бурка
- вещая Каурка

Список коней:

1. Сивка
2. Бурка
3. вещая Каурка

Можно организовать многоуровневые списки (разумно ограничиться максимум тремя уровнями вложения). Для этого после каждого <LI> следует разместить контейнеры <UL>… </UL> или <OL> … </OL> с разбивкой информации этого пункта на дополнительные подпункты.

В теге <OL> могут использоваться следующие атрибуты:

TYPE – определяет стиль оформления меток для элементов списка, подробнее ниже;

START – (только для нумерованного списка <OL>) задается всегда числом (по умолчанию 1). Число, указанное в атрибуте START, показывает порядковый номер элемента в ряду знаков, заданных в TYPE. С этого элемента следует начинать нумерацию пунктов списка.

VALUE – (только для метки <LI> списка <OL>) устанавливает номер текущего элемента списка. Как и в предыдущем случае, метка пункта может быть нечисловой, хотя значение атрибута – всегда число.

Для нумерованного списка в зависимости от стиля нумерации значением атрибута TYPE могут быть:

1 – арабские цифры (1,2,3,...)
A – прописные буквы (A,B,C,...)
a – строчные буквы (a, b, c,…)
I – римские цифры прописные (I,II,III,...)

i – римские цифры строчные (i,ii,iii,...)

По умолчанию значения TYPE устанавливаются в зависимости от степени вложенности списка.

1.9. Каскадные таблицы стилей

При форматировании шрифта возможности тегов, приведенных в п.п. 1.5–1.8, позволяют воспроизвести только основные изобразительные эффекты, улучшающие внешний вид текста. Для того чтобы расширить круг этих возможностей и сделать их использование более удобным в HTML-стандарт добавлено понятие *Каскадная таблица стилей* (CSS, Cascading Style Sheet). Стили, сформированные ее средствами, воспроизводят большинство возможностей оформления текста, предлагаемых современными текстовыми редакторами.

По способам применения стили делятся на три группы:

1) для отдельного фрагмента документа (Inline Style);

2) встроенные в документ при помощи контейнерного тега <STYLE> (Embedding Style);

3) оформленные в отдельном файле с расширением .CSS, который можно использовать в разных документах (Linking Style). Такой способ удобен при создании сайтов, состоящих из большого количества страниц.

В данном пособии рассматриваются только два первые варианта создания стилей. С методами создания Linking Style можно ознакомиться, например, в [1, 3].

Стили первой группы (Inline Style) задаются с помощью атрибута STYLE, встроенного в начальный тег форматируемого фрагмента, по следующей форме:

STYLE=”атрибут: значение; атрибут: значение; …”

В прил., табл. 4 приведены названия основных атрибутов стиля и их возможные значения.

Пример:

<H1 STYLE=”FONT-FAMILY: verdana; FONT-SIZE: 16pt; LINE-HEIGHT: 0.5in; TEXT-ALIGN: center> заголовок </H1>

Этот контейнер устанавливает следующий стиль для слова заголовок: использовать шрифт verdana размером 16 пунктов, межстрочное

расстояние 0,5 дюйма, выравнивание по центру. Остальные установки шрифта и выравнивания сохранить по умолчанию. Для других заголовков первого уровня в этом документе будет использоваться стандартное оформление.

Если фрагмент, который следует оформить нестандартным стилем, содержит несколько тегов (например, тег заголовка и тег абзаца), этот фрагмент заключают в контейнер <DIV>. Небольшой фрагмент текста, в который входят, например, только несколько слов абзаца, выделяют контейнером <SPAN>. Для таких фрагментов атрибут STYLE с его параметрами включают в открывающий тег соответствующего контейнера.

Пример:

```
<DIV STYLE=”TEXT-INDENT: 4cm”>
<H1> Весна-Лето </H1>
  <OL>
    <SPAN STYLE=”color: green”><LI> март; <LI> апрель; <LI> май; </SPAN>
    <SPAN STYLE=”color: red”><LI> июнь; <LI> июль; <LI> август; </SPAN>
  </OL>
</DIV>
```

В этом примере заголовок и список будут сдвинуты от левого края экрана на 4 см, заголовок будет оформлен обычным шрифтом, весенние месяцы – шрифтом зеленого цвета, летние – красного.

Таблица стилей, встроенная в документ (Embedding Style), позволяет заменить стандартные значения атрибутов указанных в ней тегов на те, которые удобны Вам в текущем документе. Например, при стандартных значениях атрибутов тег <HR> будет вставлять в документ тонкую серую линию от левого до правого края экрана. Если Вы в таблице стилей зададите для этого тега другие значения атрибутов, то эта же запись <HR> без указания атрибутов будет создавать в документе линию с нужным Вам оформлением.

Инструкции с нужными стилями размещаются в контейнере <STYLE>, который включается в заголовок документа (контейнер <HEAD>). Каждая инструкция имеет вид:

```
СЕЛЕКТОР {атрибут: значение; атрибут: значение; …}
```

Здесь СЕЛЕКТОР – ключевое слово тега, для которого в этом документе изменяются стандартные параметры; список атрибутов и их возможных значений приведен в прил., табл. 6.

Пример:

```
<HEAD>
<STYLE TYPE=”text/css”>
  <!--
    H1 {font-size: 20pt; font-family: Arial: font-weight: bold; color: red}
    P { font-size: 15pt; font-family: Arial; text-indent: 1.25cm}
</STYLE>
</HEAD>
```

Комбинации символов <!-- и --> в HTML-стандарте используют для выделения пояснительных текстов, которые вводятся в текст кода, но не должны быть видны в окне браузера. Атрибут TYPE=”text/css” в совокупности с символами комментария делает стилевую информацию не видимой в окне браузера, если браузер работает с версией HTML, которая не поддерживает стили.

Можно упростить запись стилевых инструкций с помощью группировки. Если один и тот же стиль назначается нескольким тегам, в селекторе можно перечислить их через запятую и обойтись только одной инструкцией. В приведенном ниже примере можно в таблице стилей две первых инструкции заменить третьей:

```
P {font-size: 14pt;}
LI { font-size: 14pt;}
P, LI {font-size: 14pt;}
```

При перечислении атрибутов также можно использовать группировку тех из них, которые начинаются с одного и того же слова. В приведенном ниже примере обе инструкции задают один и тот же стиль:

```
P {font-size: 14pt; font-family: Arial: font-weight: bold;
text-indent: 1.25cm; text-decoration: underline}
P {font: 14pt  Arial  bold; text: 1.25cm  underline}
```

1.10. Создание таблиц

Средства HTML для создания таблиц удобно использовать не только при представлении табличных данных. Их также применяют для точного выравнивания элементов на экране или комбинирования изображений и текстов. В качестве примера рассмотрим код, который создает простейшую таблицу:

Простая таблица

Первый столбец	**Второй столбец**
Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

```
<TABLE  BORDER=1  WIDTH=200 ALIGN=CENTER>
<CAPTION> Простая таблица </CAPTION>
<TR>
  <TD>Первый столбец </TD>
  <TD> Второй столбец </TD>
</TR>
<TR>
  <TD>Ячейка 1</TD>
  <TD>Ячейка 2</TD>
</TR>
<TR>
  <TD>Ячейка 3</TD>
  <TD>Ячейка 4</TD>
</TR>
</TABLE>
```

Как видно из примера, начало и конец таблицы оформляются тегами <TABLE> и </TABLE>. Все остальные теги, создающие структуру таблицы и ее содержание, вкладываются в этот контейнер. Для того чтобы создать рамку вокруг таблицы, можно задать атрибут BORDER=n, где n – толщина рамки в пикселях. По умолчанию таблицы не имеют рамок, т.е. n = 0. Атрибут WIDTH=n%, где n – целое число, позволяет занять таблицей нужный процент от ширины экрана. Если знак процента опущен, то считается, что n задает размер таблицы в пикселях.

Сразу после тега <TABLE> можно вставить контейнер, создающий заголовок таблицы:

<CAPTION> текст заголовка </CAPTION>

Заголовок по умолчанию располагается над таблицей слева. Другое выравнивание задается атрибутом ALIGN. Допустимые значения в этом контейнере:

LEFT – значение по умолчанию, заголовок над таблицей слева;
TOP – заголовок над таблицей по центру;
RIGHT – заголовок над таблицей справа;
BOTTOM – заголовок под таблицей по центру.

Таблица набирается как стопка строк, каждая из которых организуется контейнером <TR>. Информация внутри строки разбивается на ячейки с помощью контейнеров <TH>…</TH> (Table Head) и <TD>…</TD> (Table Data). Первый свою информацию выравнивает по центру и изображает жирным шрифтом, второй – выравнивает по левому краю и использует обычный шрифт. Обычно <TH>…</TH> используют для ячеек с заголовками строк или столбцов, а <TD>…</TD> – для ячеек с данными таблицы. Можно также оставить ячейку пустой, но, чтобы она была отображена на экране, следует внести в нее &-последовательность   (см. п. 1.4).

Если таблица имеет сложную структуру и какие-то ячейки занимают несколько столбцов или строк, следует в тегах <TH> или <TD> добавить атрибуты соответственно COLSPAN=n или ROWSPAN=n, где n – количество столбцов (строк), на которые растянута данная ячейка. Например, COLSPAN=3 означает, что ячейка по ширине будет простираться на 3 колонки, ROWSPAN=2 означает, что ячейка по высоте занимает две строки таблицы.

Пример:

Фрагмент кода, создающего показанную слева таблицу:

<table>
<tr><td>1</td><td colspan="3">2</td></tr>
<tr><td rowspan="2">3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td colspan="2">8</td></tr>
<tr><td>9</td><td>10</td><td>11</td><td>12</td></tr>
</table>

```
<TABLE BORDER=1>
  <TR>
    <TH> 1 </TH> <TH COLSPAN=3> 2 </TH>
  </TR>
  <TR>
    <TH ROWSPAN=2> 3 </TH> <TH> 4 </TH>
    <TH> 5 </TH> <TH > 6 </TH>
  </TR>
  <TR>
    <TH> 7 </TH> <TH COLSPAN=2> 8 </TH>
  </TR>
  <TR>
    <TH> 9 </TH> <TH> 10 </TH> <TH> 11
```

```
</TH>
    <TH> 12 </TH>
  </TR>
</TABLE>
```

Для оформления таблицы в теги <TABLE>, <TR>, <TH>, <TD> добавляют атрибуты.

Основные атрибуты тега <TABLE>:

WIDTH – определяет ширину таблицы в пикселях или процентах, по умолчанию ширина таблицы определяется содержимым ячеек;

ALIGN – определяет расположение таблицы в документе. По умолчанию таблица прижата к левому краю страницы. Допустимые значения атрибута: LEFT, CENTER и RIGHT;

BORDER – устанавливает толщину рамки пикселях. По умолчанию – 0 (таблица рисуется без рамки);

BORDERCOLOR – устанавливает цвет рамки, по умолчанию черный. Цвет можно задавать английским словом числом (см. прил., табл. 5) или шестнадцатеричным числом;

CELLSPACING – определяет расстояние между рамками ячеек таблицы в пикселах;

CELLPADDING – определяет промежуток в пикселях между рамкой ячейки и текстом внутри нее;

BGCOLOR – устанавливает цвет фона для всей таблицы. По умолчанию белый. Цвет можно задавать английским словом или шестнадцатеричным числом (см. прил., табл. 5);

BACKGROUND – заполняет фон таблицы изображением (см. п. 1.11). Значением атрибута является URL нужного файла.

Тег <TR>, открывающий строку таблицы, может иметь такие атрибуты:

ALIGN – устанавливает горизонтальное выравнивание текста во всех ячейках строки. Может принимать значения LEFT (по умолчанию), CENTER и RIGHT;

VALIGN – устанавливает вертикальное выравнивание текста в ячейках строки. Допустимые значения: TOP (выравнивание по верхнему краю), MIDDLE (выравнивание по центру – это значение принимается по умолчанию), BOTTOM (по нижнему краю);

BGCOLOR – устанавливает цвет фона для строки (см. прил., табл. 5).

Ячейки таблицы начинаются тегами <TH> или <TD>. Для них предусмотрены следующие атрибуты:

ALIGN – устанавливает горизонтальное выравнивание текста в ячейке. Может принимать значение LEFT, CENTER и RIGHT;

VALIGN – устанавливает вертикальное выравнивание текста в ячейке. Допустимые значения: TOP, CENTER (это значение принимается по умолчанию), BOTTOM;

WIDTH – определяет ширину ячейки в пикселях;

HEIGHT – определяет высоту ячейки в пикселях;

NOWRAP – присутствие этого атрибута показывает, что текст должен размещаться в одну строку;

BGCOLOR – устанавливает цвет фона ячейки (см. табл. 5);

BACKGROUND – заполняет фон ячейки изображением. Значением атрибута является URL нужного файла.

1.11. Использование изображений

Внесение в документ изображений, как правило, существенно улучшает его внешний вид и делает его привлекательным для просмотра. Файлы рисунков можно использовать как фон для основной информации или же включать в документ как самостоятельные объекты. В Интернете обычно используют изображения в формате GIF и JPEG (файлы с расширениями .gif и .jpg). Современные браузеры понимают и другие форматы графических файлов (например, .png). Все они хорошо сжимают изображение, но в то же время достаточно хорошо сохраняют его цветовую гамму и детали.

Чтобы использовать рисунок в качестве фона, применяют атрибут BACKGROUND. Его указывают в теге <BODY> (см. п. 1.4) и табличных тегах <TABLE>, <TH>, <TD> (см. п. 1.10). Если размер рисунка больше размера площади, которую следует закрыть фоном, то берется левая верхняя часть изображения, если меньше – рисунок повторяется, пока вся отведенная под фон площадь не будет закрыта, т. е. заданное изображение используется как кафельная плитка.

Рисунок в качестве самостоятельного объекта (логотип фирмы, фотография и т. п.) вставляется тегом <IMG> (IMaGe – изображение). Обязательный атрибут – SRC (SouRCe – источник). Значением этого атрибута является URL файла с изображением.

Внимание! URL обязательно надо выделять двойными кавычками и следить, чтобы пробелы и регистры букв в названиях папок и файлов в точности соответствовали оригиналу.

Примеры:

<IMG SRC="picture.gif"> – рисунок picture.gif находится в том же каталоге, что и текущий документ;

<IMG SRC="images/picture.gif"> – из текущего каталога перейти в подкаталог images и взять файл picture.gif оттуда;

<IMG SRC="../images/picture.gif"> – подняться в родительский каталог, оттуда перейти в каталог images и взять там рисунок picture.gif;

<IMG SRC="http://www.fortun.com/business/practic/myphoto.gif"> – указание полного пути к файлу. Обычно применяется, если рисунок находится на другом сервере или используется файл с другого сайта.

Остальные атрибуты тега <IMG>, перечисленные ниже, оговаривают размеры и положение рисунка на странице, а также то, где должен располагаться поясняющий текст:

BORDER=n – рамка вокруг рисунка, где n – ее толщина в пикселях. При n = 0 (значение по умолчанию) рамка не рисуется;

WIDTH=n – задается ширина изображения в пикселях или в процентах от ширины экрана браузера (тогда после значения n следует знак %). По умолчанию – естественная ширина;

HEIGHT=n – высота изображения в пикселях или в процентах от высоты экрана. По умолчанию – естественная высота;

HSPACE=n – отступ слева и справа от картинки шириной в n пикселей (т.е. свободное пространство между рисунком и текстом или чем-то иным). По умолчанию 0;

VSPACE=n – вертикальный отступ от картинки в пикселях. По умолчанию 0;

ALIGN=BOTTOM – по умолчанию. Сопровождающий текст выравнивается по нижнему краю рисунка справа от него. Другие значения атрибута ALIGN, приводящие к выравниванию текста справа от рисунка по его высоте: TOP – по верхнему краю рисунка,

CENTER или MIDDLE – по центру рисунка. Чтобы текст обтекал рисунок, используются значения LEFT или RIGHT. Рисунок прижимается соответственно к левому или правому краю экрана, а остальное пространство рядом с ним занимает текст;

ALT="какой-то текст" – альтернативный текст. Текст, который появляется вместо картинки, если по какой-либо причине загрузка изображения не состоялась.

Пример:

<IMG SRC="picture.gif" ALT="Здесь должен быть рисунок"> – рисунок picture.gif находится в том же каталоге, что и текущий документ. В том случае, если его не удается найти или открыть, в рамку, отведенную под рисунок, будет выдан текст: Здесь должен быть рисунок

Замечание. Для более аккуратного выравнивания текста около рисунка можно использовать табличные теги без атрибута BORDER. Например, если создать таблицу, состоящую из одной строки и двух ячеек в ней, то в одну можно поместить изображение, а в другую – пояснительный текст. Другие варианты придумайте сами.

1.12. Теги гиперссылок

Гиперссылки (в дальнейшем просто ссылки) – инструмент, позволяющий связывать между собой различные документы или обеспечивать быстрый переход от одной части документа к другой. Браузер обычно выделяет ссылку цветом и подчеркиванием. Ссылка имеет указатель (видимый в браузере элемент) и адрес. Щелчок по указателю приводит на объект с этим адресом. Указателем может быть текст или изображение. Указатель часто называют якорем гиперссылки. Курсор при наведении его на якорь приобретает вид указательного пальца. По двойному щелчку на ссылке браузер включает поисковые средства и раскрывает заказанный документ либо поверх старого, либо в специально оговоренном окне.

Ссылки создаются с помощью контейнера <A>…</A> (Add – добавлять). Обязательный атрибут – HREF (Hiper REFerence – гиперссылка). При минимальном наборе атрибутов структура ссылки выглядит так:

<A HREF="URL вызываемого файла"> якорь ссылки </A>

Примеры:

<A HREF="filename.htm"> нужный документ </A> – по двойному щелчку на тексте нужный документ обозреватель раскроет документ filename.htm, находящийся в том же каталоге, что и текущий документ;

<A HREF="folder/filename.htm"> нужный документ </A> – по двойному щелчку на тексте нужный документ обозреватель раскроет документ filename.htm в подкаталоге folder текущего каталога;

<A HREF="../folder/filename.htm"> нужный документ </A> – по двойному щелчку на тексте нужный документ обозреватель из каталога, в котором расположен вызывающий документ, перейдет в родительский каталог (на уровень выше), затем – в его подкаталог folder и раскроет находящийся там документ filename.htm.

<A HREF="http://www.fortunecity.com/business/fax/339/index.htm"> нужный документ </A> – ссылка с указанием полного адреса файла на другом компьютере. По двойному щелчку на тексте нужный документ установится связь с компьютером www.fortunecity.com и раскроется документ index.htm, находящийся там по адресу /business/fax/339/.

<A HREF="filename.htm"> <IMG SRC="photo.jpg" WIDTH=50> </A> – файлы photo.jpg и filename.htm находятся в том же каталоге, что и текущий документ. По двойному щелчку на картинке photo.jpg обозреватель раскроет документ filename.htm;

Другие атрибуты тега <A>:

TITLE="поясняющий текст"– всплывающая подсказка с пояснением к ссылке;

TARGET="имя окна" – указывает, в каком окне следует раскрывать вызванный документ (подробнее см. п. 1.14).

Пример (прокомментируйте сами):

<A HREF="май/ОтчетПродажиМай.htm" TITLE="Майский отчет по продажам" > просмотр отчета </A>

Замечание. Если якорем ссылки является рисунок, вокруг него появится рамка, которая показывает, что он является ссылкой. Ширина рамки задается атрибутом BORDER в теге <IMG...>. Если рамка портит внешний вид документа, то укажите в атрибутах рисунка

BORDER=0. То, что это гиперссылка, будет видно только по форме курсора, наведенного на рисунок.

Примеры:

<A HREF="ftp://servername/directory/file.exe"> нужная программа </A> – такая ссылка по щелчку на словах нужная программа запустит протокол передачи файлов (ftp) и произведет выгрузку файла file.exe из каталога directory сервера servername на жесткий диск пользователя.

Пишите письма <A HREF="mailto://Has5@mail.ru" TITLE="Ссылка на почтовый ящик Администратора сайта"> автору </A> – в окне браузера появится текст:

Пишите письма автору

Если навести курсор на слово автору, то появится всплывающая подсказка Ссылка на почтовый ящик Администратора сайта. По двойному щелчку на слове автору будет запущена почтовая программа в режиме создания сообщений, и в поле Кому автоматически будет введен адрес Has5@mail.ru

Ссылки можно делать не только на внешнюю информацию, но и на внутренние места документа. Это облегчает просмотр больших документов. Разберем это на примере. Допустим, электронный вариант этих методических указаний создается по гипертекстовому стандарту. Тогда в начале каждого параграфа необходимо создать метку, на которую будет указывать гиперссылка.

Пример:

<A NAME="п. 1.12"> </A> или <A NAME="п. 1.12"> 1.12. </A>

В первом случае после метки должен идти заголовок параграфа вместе с номером, во втором – просто заголовок, т. е. при организации метки текст якоря в контейнере <A>…</A> необязателен. Если он вставлен, то он не подчеркивается и выглядит как обычный текст. Теперь Содержание можно оформить как список гиперссылок. Символ # означает, что дальше идет метка внутри документа. Пункт Содержания со ссылкой на п. 1.12 должен выглядеть так:

<A HREF="#п.1.12"> 1.12. Теги гиперссылок </A>

На эту же метку можно будет перейти и из любого другого места методических указаний. Например, ссылку в соответствии с п. 1.1 следует оформить так:

<A HREF="#п.1.12"> (см. п. 1.12) </A>

Если гиперссылка указывает на внешний документ, то по умолчанию в окно браузера выдается его начало. Если же нужная в нем информация расположена дальше и обозначена меткой, то можно сразу же открыть документ на этом месте, указав после имени файла метку (так же, как мы открываем книгу с закладкой).

Пример:

http://www.sarf.spb.ru/study.htm#begin

Внимание! Браузеры чувствительны к регистру ссылок. Если Вы вместо "http://webs.web.com" наберете "http://webs.Web.com", то получите нерабочую ссылку. Та же ситуация и с расширениями: если у Вас ссылка на файл "index.htm", а там находится "index.html", то результат тот же – нерабочая ссылка.

1.13. Тег бегущей строки

Анимационные эффекты на HTML-странице делают ее более привлекательной. Организация сложных эффектов требует знания основ программирования на языках написания сценариев (например, JavaScript, VBAScript, Perl). Однако самые простые динамические эффекты можно включить в нее и без написания специальных программ. Это gif-анимация, она вставляется в HTML-код тегом <IMG> как обычное изображение. Можно также ввести на страничку прямолинейное движение в горизонтальном направлении, то есть бегущую строку. Она вставляется в HTML-код контейнерным тегом <MARQUEE>. При самостоятельном написании кода (без редактора HTML-страниц) в этот контейнер можно вставлять не только текст, но и теги картинок.

Характер перемещения информации и способ оформления зоны движения определяется значениями следующих атрибутов:

BEHAVIOR – тип движения: SCROLL (движение от левого края монитора к правому), ALTERNATE (дойдя до края экрана информация двигается в обратном направлении, «качается»);

LOOP – количество повторов движения: ограничение на число повторов (1, 2, 3 …и т.д.) или INFINATE (постоянные повторы до закрытия документа, значение по умолчанию);

SCROLLAMOUNT – величина каждого шага перемещения информации в пикселях;

SCROLLDELAY – период времени в миллисекундах, через которое следует делать шаги перемещения информации;

WIDTH – ширина зоны движения в пикселях или % от ширины экрана;

HEIGHT – высота зоны движения в пикселях или % от высоты экрана;

BGCOLOR – цвет фона в зоне движения.

По умолчанию информация, размещенная в контейнере <MARQUEE>, перемещается в горизонтальном направлении справа налево на прозрачном фоне. Высота зоны движения определяется вертикальным размером вставленных в нее элементов. Скорость определяется комбинацией длины шага в 20 пикселей и времени повтора шага 80 миллисекунд.

Примеры:

<MARQUEE> <IMG SRC=фото.jpg WIDTH=15% ALIGN=middle > <FONT SIZE=7> <B> Вот так фото! </FONT> </B> </MARQUEE> – в бегущей строке со стандартным оформлением размещены изображение из текущего каталога и поясняющий текст, оформленный крупным жирным шрифтом;

<MARQUEE LOOP=1 BGCOLOR=yellow SCROLLAMOUNT=1 SCROLLDELAY=80> <IMG SRC=”фото.jpg” WIDTH=15% ALIGN=center> Вот так фото! </MARQUEE> – изображение и поясняющий текст, оформленный обычным образом, один раз медленно пройдут по экрану по желтой полосе;

<MARQUEE BEHAVIOR=alternate SCROLLDELAY=1000 SCROLLAMOUNT=1300 BGCOLOR= yellow > <<IMG SRC=фото.jpg width=15% ALIGN=middle> </MARQUEE> – изображение будет перескакивать слева направо и обратно с интервалом в одну секунду.

1.14. Разделение экрана на кадры (фреймы)

Кадры (другое название *фреймы*) – это области экрана, в каждую из которых можно поместить отдельный HTML-документ. Это

удобно в тех случаях, когда при интернет-серфинге наряду с меняющейся информацией на экране желательно сохранять некоторые элементы (меню, логотип фирмы, поисковые элементы и т. п.).

Разделение окна на кадры выполняется с помощью HTML-файла, в котором контейнер <BODY>…</BODY> заменен на <FRAMESET>… </FRAMESET>. Атрибуты этого тега определяют, как разбивается экран.

Примеры:

<FRAMESET COLS="25%,40%,35%"> – экран разбивается на три колонки, ширина которых задана в процентах от ширины окна браузера;

<FRAMESET ROWS="10%,90%"> – экран разбивается на две строки, высота которых указана в процентах от высоты окна браузера.

Размеры кадров можно задавать как в пикселях или в процентах от величины экрана, так и пропорционально друг к другу. В последнем случае вместо конкретного размера используют символ *.

Примеры:

<FRAMESET COLS="100,*,35%"> – средняя колонка занимает все пространство, оставшееся от первой и последней колонок. Первая колонка на мониторе любого размера имеет ширину 100 пикселей, последняя – 35 % от ширины окна браузера;

<FRAMESET COLS="*,2*,*"> – первая и последняя колонки должны быть одинаковыми, а средняя в два раза шире их.

Заполнением кадров управляют теги <FRAME>. Обязательный атрибут – SRC="URL документа, загружаемого в кадр".

Остальные (необязательные) атрибуты тега <FRAME>:

FRAMEBORDER=n – ширина окантовки между кадрами (можно n=0);

MARGINHEIGHT=n – размер свободного пространства в пикселях у верхнего и нижнего края кадра;

MARGINWIGHT=n – размер свободного пространства в пикселях у правого и левого края кадра;

NORESIZE – отменяет для пользователя возможность изменения размера кадра за счет перетаскивания границы кадра мышью;

SCROLLING – управляет появлением горизонтальной и вертикальной полос прокрутки в кадре. Возможные значения: YES, NO, AUTO.

По умолчанию документ, раскрываемый гиперссылкой, «накладывается» на предыдущий и закрывает его. Если экран разбит на кадры, можно оговорить, в какой из них следует направить новую информацию, и она не закроет предыдущую. Это удобно при последовательном просмотре гиперссылок, входящих в меню, выделении постоянного места для средств поиска нужной информации и ряде других случаев.

Кадрам, в которые будут направляться гиперссылки из других фреймов, следует присвоить имена. Это делается атрибутом NAME="имя кадра" в теге <FRAME>. Во всех гиперссылках, нацеленных на этот кадр, в тег <A> следует добавлять атрибут TARGET="имя кадра".

Пример:

Приведенный ниже код разделяет экран на кадры по схеме:


```
<HTML>
    <HEAD>
      <TITLE> Разбивка экрана на кадры <TITLE>
    <HEAD>
    <FRAMESET  COLS="33%,67%">
      <FRAMESET  ROWS="75,*">
        <FRAME  NAME="logo"  SRC="logl.jpg">
        <FRAME  SRC=" menu.htm">
      </FRAMESET>
      <FRAME  NAME="main"  SRC="myweb.htm">
    </FRAMESET>
</HTML>
```

Здесь экран разделен на два столбца, а первый столбец – дополнительно на две строки. Ширина столбцов задана в процентах от ширины окна браузера, а высота строк – в пикселях. Считается, что

все файлы размещены в одной папке. В кадре logo размещается изображение из файла logl.jpg, в кадрах menu и main – текстовые файлы menu.htm и myweb.htm соответственно. Теперь, чтобы гиперссылки из этих документов открывались в нужном нам кадре, следует добавить в их теги атрибут TARGET. Например, для того, чтобы гиперссылка из документа menu.htm раскрывалась не в «родном» кадре, а в правом кадре с именем main, ее следует оформить так:

<A HREF="obrazovanie.htm" TARGET="main"> образование </A>

Примечание:

Если в атрибуте TARGET указано имя, которое не было присвоено ни одному кадру, то браузер раскрывает гиперссылку в новом окне. Такой прием удобно использовать для ссылок на файлы с большим объемом информации.

2. УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

2.1. Подготовка бланка для HTML-кода

Откройте в программе Блокнот новый документ. Выполните команду Файл → Сохранить как…, вписав в поле Имя вместо стандартного имени *.txt название своего документа с расширением htm. Затем закройте документ. Если все сделано правильно, Ваш документ будет помечен в папке значком браузера. Затем снова откройте этот документ. Откроется пустое окно браузера. Выполните команду Вид → Просмотр HTML-кода. Откроется блокнот с Вашим именем, в который следует заносить код создаваемого документа и все последующие исправления в нем. После очередного цикла корректировки следует сохранять изменения в блокноте и обновлять окно браузера командой Вид → Обновить.

2.2. Форматирование шрифта

1) Подготовьте бланк для HTML-кода с названием Форматирование.htm (см. п. 2.1).

2) Введите в блокнот стандартные теги структуры (см. п. 1.5). В контейнере <BODY> … </BODY> наберите один абзац текста (3–5 строк) по какой-нибудь теме и заголовок к нему. Скопируйте его 10 раз (полученный массив текста должен располагаться на месте многоточия в этом пункте). Сохраните изменения в блокноте и обно-

вите окно браузера. Посмотрите, как браузер изображает Ваш текст без форматирования. Теги, изменяющие внешний вид Вашего текста, следует вводить в код, который набран в блокноте. После оформления очередного абзаца сохраните изменения и обновите окно браузера. Если заказанные в задании эффекты достигнуты, переходите к оформлению следующего абзаца. В противном случае ищите ошибки в написании или выборе тегов и исправляйте код.

3) Первый абзац следует отформатировать с выравниванием текста по левому краю, левым абзацным отступом 0 см, правым абзацным отступом 0 см, без вертикального интервала перед следующим абзацем. Примените шрифт "Times New Roman " с размером 12 пт. Заголовок выровняйте по левому краю и изобразите самым мелким жирным шрифтом.

4) Второй абзац следует отформатировать с выравниванием текста по правому краю без интервала между соседними абзацами. Примените шрифт "Times New Roman ", с размером 20 пт. Заголовок выровняйте по правому краю и изобразите самым крупным жирным шрифтом.

5) Третий абзац следует отформатировать с правым и левым абзацными отступами по 5 символов, и вертикальными интервалами между соседними абзацами, применить шрифт "Arial "самого большого размера с начертанием "курсив". Заголовок выровняйте по центру и изобразите самым крупным жирным шрифтом.

6) Следующий абзац следует сдвинуть по отношению к предыдущим абзацам на 10 символов влево и использовать начертания "жирный", "подчеркнутый".

7) Следующий абзац отделите от предыдущих вертикальным интервалом, выровняйте по центру, первые 7 слов изобразите разными цветами и с нарастающим размером букв, в двух следующих – вторую половину слова изобразите чуть больше и чуть меньше стандартного размера шрифта. Заголовок выровняйте по правому краю и изобразите самым крупным жирным шрифтом.

8) Заголовок следующего абзаца следует изобразить жирным цветным шрифтом и выровнять по центру. В тексте устроить «лесенку»: нечетные слова изобразить как нижние индексы, четные – как верхние.

9) В этом абзаце заголовок следует изобразить обычным цветным шрифтом самого большого размера, выровнять его по середине, использовать начертание "курсив" и отделить от текста абзаца пустой

строкой. Текст абзаца следует расположить лесенкой по 3–4 слова на каждой строке. После текста вставить в документ стандартную прямую линию.

10) После следующего абзаца следует вставить в документ короткую (около двух сантиметров) цветную линию толщиной 3–5 пунктов, выровненную по центру.

11) Последний абзац отформатируйте по своему усмотрению.

2.3. Списки

1) Подготовьте бланк для HTML-кода с названием Списки.htm (см. п. 2.1).Введите в блокнот стандартные теги структуры (см. п. 1.5). В контейнере <BODY>…</BODY> наберите один абзац текста (2–5 строк) по какой-нибудь теме. Скопируйте его 9 раз (полученный массив текста должен располагаться на месте многоточия в этом пункте). Сохраните изменения в блокноте и обновите окно браузера.

2) Оформите первые три абзаца как маркированный список со стандартными параметрами, следующие три – как стандартный нумерованный список, остальные просто оформите тегами <LI> без контейнеров <OL> или <UL>.

3) Над каждым списком внесите заголовки: МАРКИРОВАННЫЙ, НУМЕРОВАННЫЙ, БЕСХОЗНЫЙ соответственно.

4) Измените тип нумерации и вид маркеров в списках. Отметьте, можно ли внести эти изменения в последний список.

5) Перенесите заголовки двух первых списков внутрь соответствующих контейнеров перед первым пунктом (тегом <LI>). Посмотрите, как при этом изменяется оформление заголовка.

6) Создайте двухуровневый список с заголовком РАСПИСАНИЕ ЗАНЯТИЙ. Для этого сначала создайте маркированный список дней недели. Содержанием каждого пункта сделайте вложенный нумерованный список с заголовком ПАРЫ. Красиво отформатируйте заголовки.

2.4. Таблицы и управление цветом

1) Подготовьте бланк для HTML-кода с названием Таблицы.htm (см. п. 2.1). Введите в блокнот стандартные теги структуры (см. п. 1.5). Сохраните изменения в блокноте и обновите окно браузера.

2) В контейнер <BODY>…</BODY> введите следующий текст:

```
<CENTER>
    <FONT COLOR="#FF0000" SIZE=+4>
      Управление цветом
    </FONT>
</CENTER>
  <B>
    <FONT COLOR="#808080" SIZE=+2>
      Так может выглядеть компьютерная радуга:
    </FONT>
  </B>
  <P><P>
  <TABLE BORDER=1 WIDTH="80%" BORDERCOLOR=#FFFFFF>
    <TR><TD BGCOLOR="#FF0000">Красный</TD></TR>
    <TR><TD BGCOLOR="#FF7800">Оранжевый</TD></TR>
    <TR><TD BGCOLOR="#FFFF00">Желтый</TD></TR>
    <TR><TD BGCOLOR="#00FF00">Зеленый</TD></TR>
    <TR><TD BGCOLOR="#43D8FB">Голубой</TD></TR>
    <TR><TD BGCOLOR="#0000FF">Синий</TD></TR>
    <TR><TD BGCOLOR="#8000C0">Фиолетовый</TD></TR>
  </TABLE>
```

3) Посмотрите, как выглядит этот код в браузере. Найдите коды базовых цветов системы RGB, коды серого и белого цвета, остальные коды. Замените серый цвет надписи более интенсивным серым, цвет разлиновки таблицы – черным.

4) Замените некоторые коды текстовыми названиями цвета.

5) В каждой строке добавьте еще по одной ячейке так, чтобы во втором столбце цвета радуги шли в обратном порядке (это удобно делать копированием уже существующих контейнеров ячеек).

6) Добавьте третий столбец и в его ячейках потренируйтесь с составлением кодов других цветов.

7) Создайте коды по макетам представленных таблиц. Способ форматирования каждой таблицы указан рядом с нею.

Залейте одинаковым цветом ячейки, относящиеся к одной и той же строке (1/2/3/4, 5/6/7, 8/9/10, 11/12). Для разных строк используйте разные цвета.

Вариант 1

<table>
<tr><td colspan="4">Вариант 1</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td colspan="2">5</td><td rowspan="2">6</td><td>7</td></tr>
<tr><td rowspan="2">8</td><td>9</td><td>10</td></tr>
<tr><td colspan="2">11</td><td>12</td></tr>
</table>

Закажите цветную разлиновку, просвет между ячейками, ширину таблицы 40 % от ширины экрана, выравнивание таблицы по горизонтали по центру.

Вариант 2

<table>
<tr><td rowspan="3">1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td colspan="3">5</td></tr>
<tr><td>6</td><td colspan="2">7</td></tr>
<tr><td colspan="3">8</td><td>9</td></tr>
</table>

Закажите красный шрифт для левого списка и синий – для правого, отсутствие разлиновки, ширина таблицы 50 %, выравнивание в ячейках по центру.

Вариант 3

Здесь расположите список друзей женского пола	Здесь расположите список друзей мужского пола

Закажите длину и ширину таблицы 100 %, отсутствие разлиновки. Текст оформите самым крупным жирным шрифтом и выровняйте по центру. В ячейке используйте заливку. Посмотрите, как он будет размещаться при изменении размеров окна браузера.

Вариант 4

Привет!

2.5. Использование изображений

1) Подготовьте бланк для HTML-кода с названием Картинки.htm (см. п. 2.1). Введите в блокнот стандартные теги структуры (см. п. 1.5). В контейнере <BODY>…</BODY> наберите один абзац текста (5–10 строк) по какой-нибудь теме. Скопируйте его 3 раза (полученный массив текста должен располагаться на месте многоточия в этом пункте). Сохраните изменения в блокноте и обновите окно браузера. Найдите в Вашем компьютере несколько файлов с изображениями людей и пейзажей разной тональности в JPG- и GIF-

формате, скопируйте из Интернета несколько анимационных GIF-изображений. Скопируйте их в ту папку, в которой находится файл Картинки.htm.

2) Сделайте первый пейзаж фоном рабочего окна браузера, изображения людей вставьте внутрь текстов первых трех абзацев Вашего текста. Если изображения очень большие, сожмите их соответствующими атрибутами так, чтобы каждое из них занимало не больше 10 % экрана. Посмотрите, как компонуется текст и изображение по умолчанию.

3) Создайте вложенную папку и перенесите туда (а не скопируйте) файл какой-нибудь картинки, которая использовалась в предыдущем задании. Файл другой картинки перенесите в папку верхнего уровня. Обновите окно браузера. Измените пути к файлу в тегах картинок так, чтобы после обновления они снова появились на экране.

4) Закажите для каждой картинки разные варианты обтекания текстом (с левой, правой стороны от картинки, с разрыванием текста). Оформите рамку и поля вокруг них, всплывающие подсказки по содержанию.

5) Создайте таблицу из двух строк и двух столбцов. В каждую ячейку поместите по тегу с изображениями пейзажей или GIF-анимацией. Задайте одинаковые размеры для всех картинок. Сравните эффекты при задании размеров в пикселях и процентах. Добавьте в каждую ячейку поясняющий текст по каждой картинке. Сравните внешний вид компоновки информации при наличии рамки у таблицы и без нее.

2.6. Гиперссылки и метки

1) Вставьте в каждую из страничек, созданных в п. п. 2.2–2.5 гиперссылки на другие страницы из этой группы так, чтобы браузер по вашим ссылкам мог попасть в каждую из них, начав просмотр с любой. В качестве якорей ссылок можно использовать названия файлов без расширений или какой-нибудь другой текст, указывающий на тип информации в вызываемом файле.

2) Внесите изменения в таблицу, созданную в задании 5 файла Картинки.htm: каждую картинку уменьшите до размеров марки (примерный размер по горизонтали 40–60 пикселей) и сделайте ее якорем гиперссылки, раскрывающей эту картинку в ее естественном размере.

3) Внесите изменения в файл Форматирование.htm: в заголовок каждого абзаца введите его номер. Пять первых абзацев пометьте невидимыми метками с номерами абзацев, остальные – видимыми. Добавьте невидимую метку в конце файла. В начало файла введите ссылки на какой-нибудь промежуточный абзац и на конечную метку. В конце файла – ссылку на первый абзац. Посмотрите, как выглядят Ваши метки в окне браузера. Проверьте, как работают внесенные гиперссылки.

4) Внесите в файлы Списки.htm, Таблицы.htm и Картинки.htm гиперссылки с метками на последние абзацы файла Форматирование.htm. Проверьте, как они работают.

2.7. Бегущие строки

1) Подготовьте бланк для HTML-кода с названием Бегущие строки.htm (см. п. 2.1). Введите в блокнот стандартные теги структуры (см. п. 1.5). Сохраните изменения в блокноте и обновите окно браузера.

2) В контейнер <BODY>…</BODY> введите контейнер бегущей строки с каким-нибудь текстом. Посмотрите, как выглядит текст в окне браузера. Красиво отформатируйте текст. Поэкспериментируйте со значениями атрибутов бегущей строки (скорость перемещения информации, высота, заливка, длина полосы и тип движения).

3) Добавьте к двигающемуся тексту тег какой-нибудь картинки. Подберите атрибут высоты картинки так, чтобы зона движения занимала примерно 10 % от высоты экрана.

4) Создайте «гонки картинок». Для этого сделайте 4 копии бегущей строки и задайте в них длину полосы движения 100 % от ширины экрана, а заливку и скорость движения разными. Вместо текста вставьте в каждую копию разные картинки (желательно с GIF-анимацией). Посмотрите, как это будет выглядеть в окне браузера.

5) Создайте таблицу из двух строк и двух столбцов с атрибутом ширины равным 80 %. В каждую ячейку поместите по тегу бегущей строки с изображениями пейзажей или GIF-анимацией. Задайте небольшие одинаковые размеры для всех картинок. Посмотрите, как этот код будет выглядеть в окне браузера.

2.8. Фреймы

1) Создайте пять бланков HTML-документов с названиями ФИО.htm, Меню.htm, Параметры.htm, Друзья.htm, Увлечения.htm, Фотосессия.htm (для уменьшения объема работы их можно размножить копированием). В первый документ введите свою фамилию и фотографию, красиво оформите эту информацию и поместите ее в бегущей строке. В документ Параметры.htm поместите основные физические характеристики личности: возраст, рост, вес, цвет волос и т. п. В документы Друзья.htm и Увлечения.htm вставьте списки с кратким описанием своих друзей и увлечений соответственно. Оформите эти файлы разными вариантами заливки, списков, шрифта. В документ Меню.htm поместите гиперссылки на файлы Параметры.htm, Друзья.htm, Увлечения.htm, расположенные в одной строке с несколькими пробелами между ними (см. п. 1.4, 1.7). В файле Фотосессия.htm можно создать таблицу, в ячейки которой вставить серию своих фотографий.

2) Подготовьте бланк для HTML-кода с названием Index.htm (см. п. 2.1). Введите в блокнот стандартные теги структуры (см. п. 1.5), заменив контейнер <BODY>…</BODY> на контейнер <FRAMESET>…</ FRAMESET > с атрибутами и тегами <FRAME>, которые заполнят окно браузера по следующей схеме:

Сохраните изменения в блокноте и обновите окно браузера.

ФИО.htm	100 пкс,Меню.htm	В два раза шире, чем первый столбец, Фотосессия.htm

Посмотрите, как будут размещаться и раскрываться ссылки в меню при этой схеме организации фреймов.

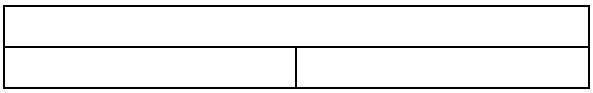
Измените атрибуты в теге <FRAMESET> так, чтобы место, отведенное в окне браузера для этих файлов, разделялось горизонтальными линиями в следующих пропорциях: верхний кадр – 150 пкс (файл ФИО.htm), второй – 30 пкс (файл Меню.htm), третий – все оставшееся пространство. Посмотрите, как будут размещаться и раскрываться ссылки в меню при этой схеме организации фреймов.

3) Присвойте имя нижнему фрейму и нацельте ссылки на файлы Параметры.htm и Друзья.htm в этот фрейм, а ссылку на Увлечения.htm – в отдельное окно браузера. Добавьте в каждый

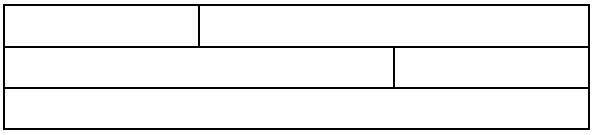
из вызываемых файлов ссылки на файл Фотосессия.htm, который должен раскрываться в нижнем фрейме.

4) Создайте файлы Index1.htm, Index2.htm, Index3.htm, Index4.htm, которые будут располагать один и тот же файл в разных фреймах, созданных по приведенным ниже макетам.

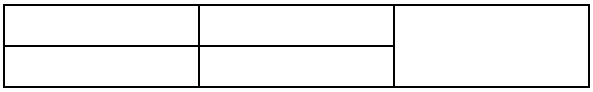
Index1.htm



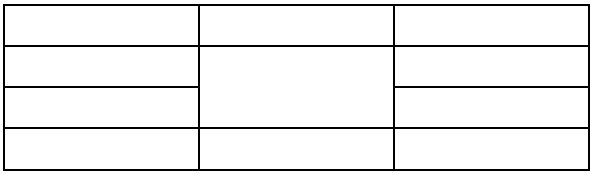
Index2.htm



Index3.htm



Index4.htm



СПИСОК ЛИТЕРАТУРЫ

1. **Д. О'Доннел**, **Э. Лэдд**. Microsoft®Internet Explorer в подлиннике. – СПб.: BHV-Санкт-Петербург, 1998. – 720 с.

2. **Долженков В.А.**, **Колесников Ю.В**. Самоучитель Microsoft Excel 2002. – СПб.: БХВ-Петербург, 2002. – 432 с.

3. http://html.manual.ru/book/html/body/hyperlinks/a.php

4. http://www.postroika.ru/html/

5. http://ru.html.net/tutorials/html/

ПРИЛОЖЕНИЕ

Таблица 1

Основные теги форматирования и компоновки текста

Теги	Действие
<B>…</B>	**Полужирный шрифт**
<I>…</I>	*Курсив*
<BIG>…</BIG>	Более крупный шрифт
<SMALL>…</SMALL>	Более мелкий шрифт
_…	Нижний индекс
[…]	Верхний индекс
<H1>…</H1>	Самый крупный размер заголовка
<H2>…</H2>, … <H5>…</H5>	**Промежуточные размеры заголовков**
<H6>…</H6>	**Самый мелкий размер заголовка**
 	Переход на новую строку
<P>…</P>	Новый абзац
<HR>	Горизонтальная линия
<PRE>…</PRE>	Компоновка текста, как в коде
<OL>…</OL>	Нумерованный список
<UL>…</UL>	Маркированный список
<LI>	Элемент списка
<TABLE>…</TABLE>	Таблица
<CAPTION>…</ CAPTION>	Заголовок таблицы
<TR>…</TR>	Строка таблицы
<TH>…</TH>	Ячейка заголовка
<TD>…</TD>	Ячейка данных
<FONT>…</FONT>	Цвет, тип и 7 стандартных размеров шрифта от самого мелкого до самого крупного

Таблица 2

Теги внесения ссылок на дополнительную информацию

Теги	Действие
<A HREF="URL файла-ссылки">…</A>	Ссылка на текстовый файл или рисунок
<A NAME="#имя метки">…</A>	Установка метки внутри файла
<IMG SRC="URL файла-изображения">	Вставка изображений

Таблица 3

Основные атрибуты (первое из перечисленных значений – значение по умолчанию)

Атрибут	Возможные значения	Действие атрибута	В каких тегах используется
COLOR=	1) Текстовое название цвета: GRAY (серый), AQUA (аквамарин), BLACK (черный), BLUE (синий), FUCHSIA (яркий пурпурно-красный), GREEN (зеленый), LIME (зеленоватый), MAROON (темно-бордовый), NAVY (темно-синий), OLIVE (оливковый), PURPLE (пурпурный), RED (красный), SILVER (серебристый), TEAL, YELLOW (желтый), WHITE (белый). Возможны также те же слова с приставками LIGHT и DARK, например LIGHTGREEN (светло-зеленый), DARKBLUE (темно-голубой) 2) Шестнадцатеричный код в системе RGB	Задает цвет линий и шрифта в тексте или таблице	<HR>, <FONT>
BGCOLOR=		Задает цвет фона	<TABLE>, <TR>, <TH>, <BODY>
BORDERCOLOR=		Задает цвет внешнего контура таблицы	<TABLE>
TEXT=		Задает цвет шрифта в документе в целом	<BODY>
LINK=, VLINK=, ALINK=		Цвета соответственно непосещенных, посещенных и активных ссылок	<BODY>

Продолжение табл. 3

Атрибут	Возможные значения	Действие атрибута	В каких тегах используется
BACKGROUND=	"URL" файла с изображением для фона	Создает фон-картинку	<TABLE>, <TH>, <TD>, <BODY>
BORDER=	Целое число без размерности	Задает толщину окантовки для изображения или таблицы	<IMG>, <TABLE>
ALIGN=	LEFT, CENTER, RIGHT	Горизонтальное выравнивание текстового фрагмента или таблицы в целом	<P>, <H1>, <H2>, …, <H6>, <TABLE>, <HR>, <TH>, <TD>
ALIGN=	BOTTOM, TOP	Размещение заголовка над или под таблицей	<CAPTION>
ALIGN=	BOTTOM, LEFT, RIGHT, TOP, MIDDLE	Размещение текстового пояснения к картинке	<IMG>
VALIGN=	MIDDLE, TOP, BOTTOM	Вертикальное выравнивание фрагмента	<TABLE>, <TH>, <TD>
WIDTH=	Целое число без размерности или со знаком %	Длина/высота фрагмента в пикселах или в процентах от ширины/высоты окна. Для всех ячеек, находящихся в строке/столбце, используется максимальное значение из заданных в ее/его ячейках	<TABLE>, <HR>, <TH>, <TD>, <IMG>
HEIGHT=			

Окончание табл. 3

Атрибут	Возможные значения	Действие атрибута	В каких тегах используется
SIZE=	Целое число без размерности (по умолчанию 1)	Толщина линии, размер шрифта	<HR>, <FONT>
TYPE=	1, A, a, i, I	Тип нумерации элементов упорядоченного списка	<OL>
START=	Номер первого элемента в выбранном типе нумерации		

Таблица 4

Атрибуты, задающие стиль

Атрибут	Действие атрибута и возможные значения
FONT-FAMILY	Указывает название шрифта. Если он недоступен браузеру, берется шрифт по умолчанию.
FONT-SIZE	Устанавливает размер шрифта в пунктах (pt), дюймах (in), сантиметрах (cm), пикселях (px).
FONT-WEIGHT	Регулирует начертание шрифта: NORMAL (нормальный), BOLD (полужирный).
FONT-STYLE	Применение курсива. Единственное значение – ITALIC (курсив).
TEXT-DECORATION	Применение текстовых эффектов: NONE (отсутствие эффектов, underline (подчеркивание), ITALIC (курсив), inethrough (перечеркивание).
TEXT-ALIGN	Горизонтальное выравнивание текста: LEFT (левое), RIGHT (правое), CENTER (по центру).
VERTICAL-ALIGN	Вертикальное выравнивание текста: BASELINE (основание), SUB (нижний индекс), SUPER (верхний индекс), TOP (верх), TEXT-TOP (верх текста), MIDDLE (середина), BOTTOM (низ), TEXT-BOTTOM (низ текста), процент от значения, заданного атрибутом LINE-HEIGHT.
TEXT-INDENT	Устанавливает размер отступа в пунктах (pt), дюймах (in), сантиметрах (cm), пикселях (px).

Окончание табл. 4

Атрибут	Действие атрибута и возможные значения
TEXT-TRANSFORM	Управляет регистром букв: CAPITALIZE (первая буква слова заглавная). UPPERCASE (верхний регистр), LOWER (нижний регистр).
COLOR	Устанавливает цвет шрифта: либо текстовое название (см. табл. 5), либо шестнадцатеричный код в системе RGB.
MARGIN-LEFT	Устанавливает размер левого поля в пунктах (pt), дюймах (in), сантиметрах (cm), пикселях (px).
MARGIN-RIGHT	Устанавливает размер правого поля в пунктах (pt), дюймах (in), сантиметрах (cm), пикселях (px).
MARGIN-TOP	Устанавливает размер верхнего поля в пунктах (pt), дюймах (in), сантиметрах (cm), пикселях (px).
MARGIN-BOTTOM	Устанавливает размер нижнего поля в пунктах (pt), дюймах (in), сантиметрах (cm), пикселях (px).
LINE-HEIGHT	Устанавливает межстрочный интервал в пунктах (pt), дюймах (in), сантиметрах (cm), пикселях (px).
BACKGROUND	Устанавливает фоновое изображение: URL изображения, или цвет фона: либо текстовое название цвета (см. табл. 5), либо шестнадцатеричный код в системе RGB.

СОДЕРЖАНИЕ

Петрунина Елена Борисовна
Селина Елена Георгиевна

ОСНОВЫ HTML

Учебно-методическое пособие

Ответственный редактор
Т.Г. Смирнова

Редактор
Л.Г. Лебедева

Компьютерная верстка
Н.В. Гуральник

Дизайн обложки
Н.А. Потехина

Подписано в печать 05.02.2013. Формат 60×84 1/16
Усл. печ. л. 3,02. Печ. л. 3,25. Уч.-изд. л. 3,0
Тираж 200 экз. Заказ № С 4

НИУ ИТМО. 197101, Санкт-Петербург, Кронверкский пр., 49
ИИК ИХиБТ. 191002, Санкт-Петербург, ул. Ломоносова, 9