

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ

Ек. А. Васенина

ОСНОВЫ ЯЗЫКА HTML

Учебно-методическое пособие
для студентов прикладных и экономических специальностей

Киров
2005

ББК 32.973.26-018.1я73
В19

Печатается по решению редакционно-издательского совета Вятского государственного гуманитарного университета

Рецензенты: И. А. Бабушкина, кандидат педагогических наук, доцент
М. В. Петухова, кандидат педагогических наук, доцент

В19 Васенина, Ек. А. Основы языка HTML: Учебно-методическое пособие для студентов прикладных и экономических специальностей [Текст] / Ек. А. Васенина. – Киров: Изд-во ВятГГУ, 2005. - 106 с.

В пособии изложены основные направления использования HTML: форматирование текста, работа с графикой и таблицами, реализация гипертекстовых связей между web-страничками, создание форм, использование фреймов, использование каскадных таблиц стилей, слои и позиционирование. В каждой главе теоретический материал сопровождается упражнениями и заданиями, позволяющими закрепить изученный материал. В конце главы приводится список теоретических вопросов для повторения и одно или несколько контрольных заданий, которые позволяют оценить качество полученных знаний по пройденной главе. Проектное задание в конце пособия дает возможность обобщить и структурировать полученные знания и перенести приобретенные умения в новую, незнакомую среду. Выполнение проектного задания может быть «сквозным», т. е. осуществляться в течение всего курса.

Пособие написано по материалам практических занятий со студентами Вятского государственного гуманитарного университета.

Издание можно рекомендовать для индивидуального обучения и для использования в качестве учебно-методического пособия в гуманитарных вузах и средних учебных заведениях различного типа.

© Вятский государственный гуманитарный университет (ВятГГУ), 2005
© Васенина Ек. А., 2005

Введение

Вся информация в сети Интернет хранится в виде страниц с гипертекстовыми связями – ссылками на другие страницы. Такие страницы еще называют Интернет-страницами, web-страницами, HTML-страницами или HTML-документами.

Web-страничка - это обычный текстовый файл в соответствующей кодировке, с расширением *.htm, *.html, *.dhtml, *.shtml¹ и т. д. В нем описывается вся страничка с помощью специального языка - HTML (*hypertext mark-up language* – гипертекстовый язык разметки документов). В основе HTML лежат тэги. Тэги – это управляющие элементы, которые определяют параметры отображения той или иной части HTML-документа: размер, цвет и начертание символов, размеры рисунка, выравнивание абзацев, расположение объектов на странице и т. д.

Для просмотра готовых web-страниц используются специальные программы – браузеры. Браузер (встречаются такие термины, как броузер или браузер) предназначен для просмотра web-страничек, которые находятся в сети Интернет или на вашем компьютере. Существует большое количество различных браузеров. На момент написания данного пособия самые распространенные и наиболее часто используемые - это Internet Explorer (компания Microsoft), Mozilla Firefox (компания Mozilla) и Netscape Navigator (компания Netscape). Браузеры являются бесплатными программами и доступны для скачивания на официальных сайтах компаний Microsoft, Mozilla и Netscape. Последней версией Internet Explorer'a является 6-я версия² (в том числе русская 6-я версия), Mozilla Firefox 1-й версии и Netscape 7-й версии (в том числе русская 7-я версия).

Когда вы в своем браузере загружаете web-страничку, браузер выполняет команды, записанные на языке HTML, и, подчиняясь им, выводит на экран страничку. Различные браузеры могут поддерживать или нет различное форматирование, поэтому внешний вид web-странички зависит от браузера, в котором она просматривается.

Часто, говоря о web-строительстве, мы ведем речь о сайтах и страничках. Чем же web-сайт отличается от web-странички? Web-страничка (иногда еще говорят, HTML-страничка) - это текстовый файл, написанный на языке HTML в определенной кодировке и с определенным расширением. Когда говорят о web-страничке, подразумевают один, два или три таких файла, которые могут быть связаны друг с другом или нет, иметь ссылки на другие ресурсы Интернета или нет.

А web-сайт содержит много (больше трех) файлов в формате HTML, обязательно связанных между собой гиперссылками. Как правило, все эти файлы оформлены в едином стиле, имеют одно и то же расширение и посвящены одной или нескольким связанным между собой темам.

¹ В данном пособии все HTML-документы для определенности имеют расширение *.html.

² В данном пособии при выполнении упражнений была использована 6-я версия Internet Explorer.

Создание сайта или странички начинается с проектирования. В процессе проектирования разрабатывается техническое задание на создание сайта, где указывается информация, которую требуется учитывать при непосредственном создании сайта: перечисление разделов сайта, макеты страниц, основные параметры форматирования страниц и замечания по содержанию сайта.

Затем начинается техническая реализация сайта: создание HTML-страничек, форматирование и т. д.

Создавать web-странички можно как вручную - путем написания кода на HTML (в любом текстовом редакторе, чаще всего в стандартном приложении Windows «Блокнот»), так и с помощью специальных визуальных редакторов («Microsoft FrontPage», «Macromedia Dreamweaver»).

Каждый способ имеет ряд плюсов и минусов.

Визуальные редакторы позволяют создавать web-странички без особых знаний HTML. Кроме того, при создании странички более сложной структуры, гораздо удобнее «видеть» ее.

Недостатком работы с программами-редакторами является тот факт, что они создают не такой оптимальный HTML-код, как опытный web-мастер, да и встраивать многие элементы возможно, только непосредственно редактируя код странички.

Создавая странички вручную, вы получаете страничку с меньшим количеством кода. Следовательно, ваша страничка будет быстрее загружаться из Интернета (а это довольно важно для профессиональных и любительских сайтов). Кроме того, создавая страничку через HTML-код, вы сможете использовать многие дополнительные возможности, которые невозможно реализовать с помощью редакторов, например DHTML.

Как было сказано выше, чтобы создавать страничку вручную, нужны знания основ HTML. Поэтому, если вы собираетесь делать более или менее серьезный сайт, то знание HTML просто необходимо.

Глава 1. Структура HTML-документа

Просмотр кода готовой web-страницы

Для того чтобы посмотреть HTML-код страницы, на которой вы сейчас находитесь, нужно в меню браузера выбрать команду **Вид/Просмотр HTML-кода** или щелкнуть правой кнопкой мыши по рабочему полю страницы и в контекстном меню выбрать пункт **Просмотр HTML-кода** (рис. 1.1).

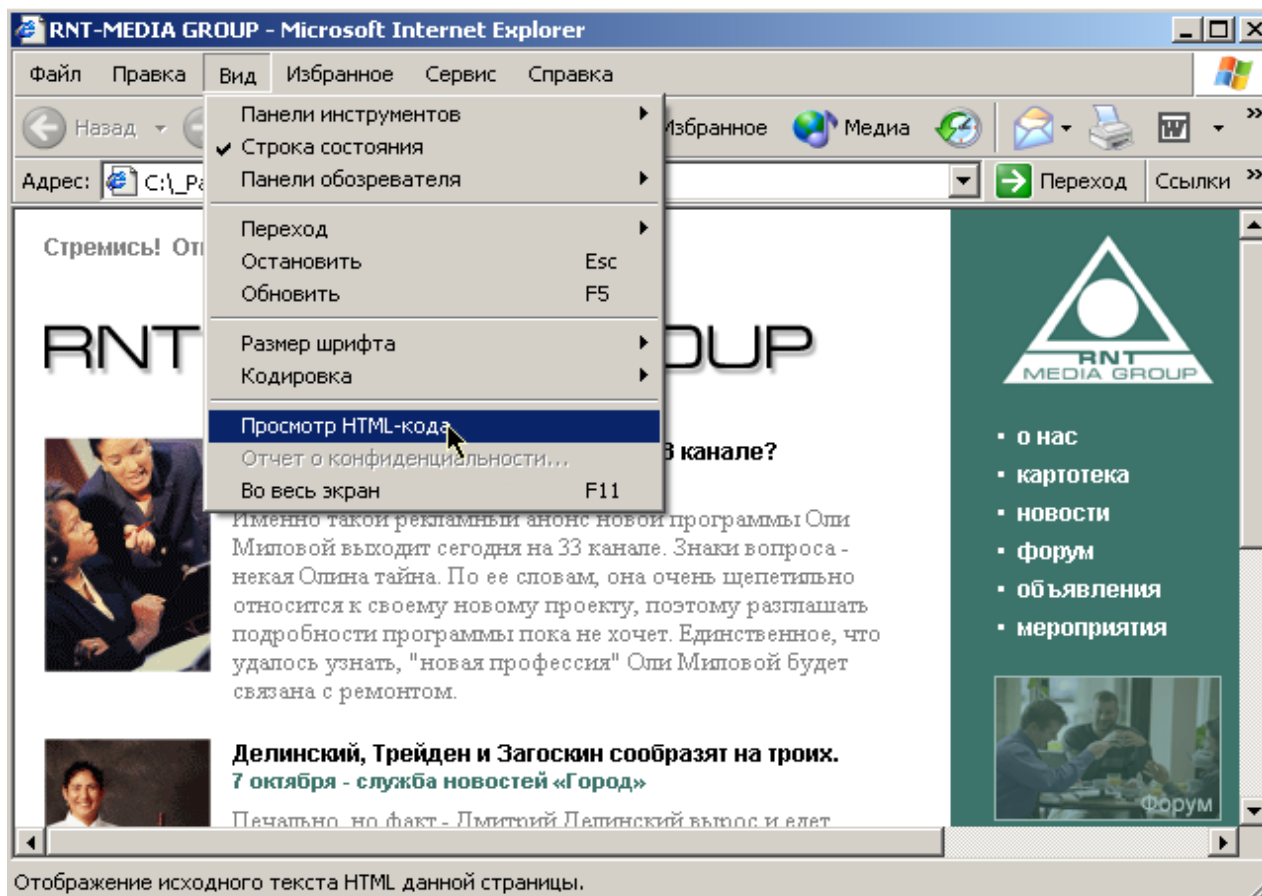


Рис. 1.1. Просмотр HTML-кода через команду главного меню

Упражнение 1.1

Загрузите любую страницу из Интернета и посмотрите код этой странички.

Структура HTML-документа

HTML-документ (или web-страничка) создается в любом текстовом редакторе и представляет собой простой текстовый документ, в который вставлены флаги разметки – «тэги» (markup tags). Все тэги записываются в угловых скобках - < >. Существуют тэги парные (имеют открывающийся элемент, заключенный в угловых скобках < >, и закрывающийся, заключенный между </ и >) и непарные.

Любой HTML-документ должен начинаться открывающим тэгом **<HTML>** и заканчиваться закрывающим тэгом **</HTML>**. Эти тэги показывают, что находящиеся между ними строки представляют единый HTML-документ. Без этих тэгов браузер или другая программа просмотра не в состоянии идентифицировать HTML-формат документа и правильно его интерпретировать.

Примечание. Хотя большинство современных браузеров могут распознать документ и не содержащий тэгов **<HTML>** и **</HTML>**, все же их употребление крайне желательно, так как некоторые программы с большой долей вероятности примут HTML-документ без открывающего и закрывающего тэгов за обычный текстовый документ. В этом случае в окне программы просмотра откроется код HTML-документа, который мы обычно видим, выполнив команду **Вид/Просмотра HTML кода**.

HTML-документ состоит из двух частей: голова (HEAD) и тело (BODY).

Голова HTML-документа заключается между парными тэгами **<HEAD>** и **</HEAD>**. В нем указывается заголовок (название) создаваемой web-странички (при просмотре в окне браузера оно отображается в заголовке окна браузера) и определяется кодировка страницы.

Заголовок странички располагается между парными тэгами **<TITLE>** и **</TITLE>**. Например,

<TITLE>Моя страничка</TITLE>.

Кодировка HTML-документа используется для отображения на странице символов кириллицы. Самые распространенные кодировки – это «Windows-1251» (стандарт Windows) и «KOI8-r» (стандарт UNIX и некоторых других систем). Соответственно символы кириллицы могут быть закодированы в одном из этих стандартов.

Если кодировку не определить в коде странички, то браузер будет отображать ее в кодировке, установленной по умолчанию.

Для определения кодировки странички браузером между тэгами **<HEAD>** и **</HEAD>** нужно вставить одну из следующих строчек:

<META http-equiv="Content-Type" content="text/html; charset=windows-1251">

- для Windows-1251 кодировки. В этой кодировке вы можете писать код странички в редакторе «Блокнот», используя как русский, так и английский язык; эта кодировка наиболее предпочтительна.

<META http-equiv="Content-Type" content="text/html; charset=koi8-r">

- для KOI8-r кодировки. Как правило, эта кодировка используется визуальными HTML-редакторами, открыв документ этой кодировки в редакторе «Блокнот», вы увидите лишь непонятные символы, однако

некоторые серверы работают лишь с этой кодировкой. В этом случае вам придется использовать специальные перекодировщики).

Тело HTML-документа находится между парными тэгами **<BODY>** и **</BODY>**. Здесь располагается информация, которую мы видим в окне браузера при просмотре странички: текст, кнопки, рисунки, списки и т. д.

Таким образом, структуру HTML-документа можно представить в виде схемы (рис. 1.2).

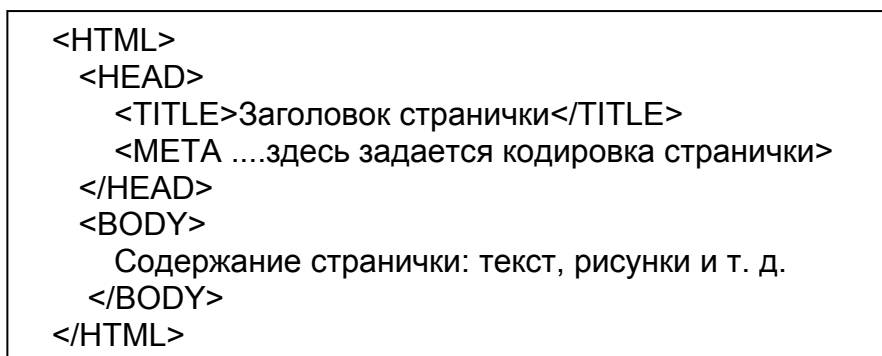


Рис. 1.2. Структура HTML-документа

Упражнение 1.2

Создайте HTML-документ *main.html* с заголовком «Страничка Ивана Иванова» со следующим содержанием: «Здравствуйте, это моя первая страничка!».

Для этого:

- В любом текстовом редакторе (например, «Блокнот») создайте новый документ.
- Внесите в документ следующий текст:

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
<TITLE> Страничка Ивана Иванова</TITLE>
</HEAD>
<BODY>
Здравствуйте, это моя первая страничка!
</BODY>
</HTML>
```

- Сохраните этот файл в свою папку под именем *main.html*. Для этого в окне сохранения документа в поле **Тип файла** выберите *Все файлы*.
- Просмотрите страничку в браузере: просто щелкните по вашему файлу двойным щелчком левой кнопкой мыши. Если расширение файла задано правильно, то автоматически файл откроется в окне браузера (см. рис. 1.3).

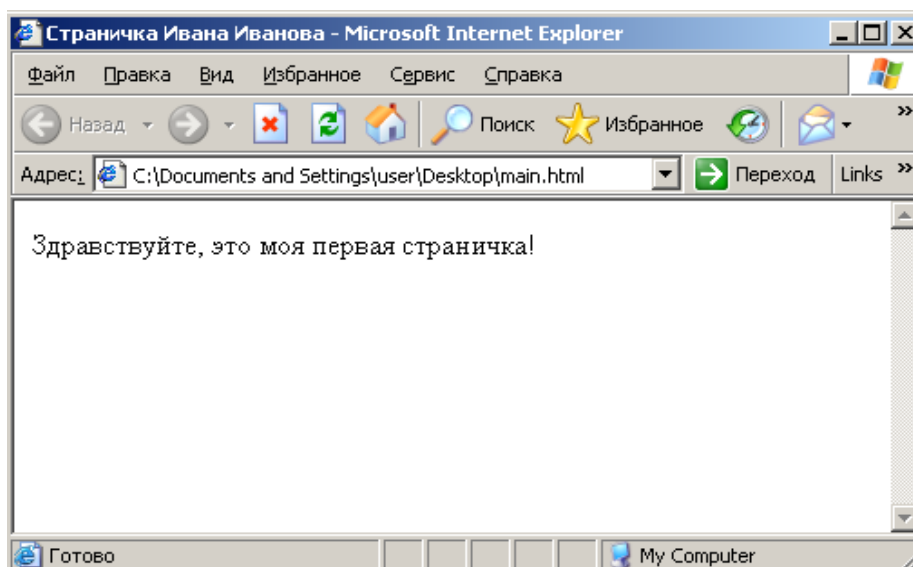


Рис. 1.3. Простейший HTML-документ

Упражнение 1.3

Измените HTML-документ *main.html*: в заголовок внесите свою фамилию и имя.

Для этого:

- Откройте HTML-код странички:
 - 1-й способ. Через меню браузера – **Вид/Просмотра HTML-кода**.
 - 2-й способ. Через меню текстового редактора – **Файл/Открыть**, поле **Тип файла** выбрать *Все файлы* и выбрать нужный HTML-файл.
 - 3-й способ. Через контекстное меню html-файла – щелкнув правой кнопкой мыши по значку файла, вызвать контекстное меню и в пункте **Открыть с помощью...** выбрать *Блокнот*.
- В разделе `<TITLE> </TITLE>` внесите свою фамилию и имя.
- Сохраните изменения.
- Просмотрите страничку в окне браузера.

Вопросы для повторения

1. С помощью какой команды главного меню браузера Internet Explorer можно просмотреть HTML-код web-страницы?
2. Чем HTML-документ отличается от простого текстового документа?
3. На какие 2 части можно логически разделить обычный HTML-документ?
4. Какие тэги задают заголовок HTML-документа? Какая информация может содержаться в заголовке HTML-документа?
5. Какие тэги задают тело HTML-документа? Какая информация содержится в теле HTML-документа?
6. Что такое кодировка?
7. Какое расширение может иметь HTML-документ?

Контрольное задание

Создайте HTML-документ в кодировке KOI8-г с названием странички «Вторая страничка ... (здесь впишите свою фамилию и имя)» и с текстом «Всем привет! Мы изучили структуру HTML-документа! Эта страничка создана в кодировке KOI8-г».

Глава 2. Форматирование HTML-документа

К форматированию документа можно отнести изменение внешнего вида документа: форматирование страницы, разбиение текста на абзацы, выделение заголовков, форматирование символов и др.

Настройка внешнего вида страницы

Задавая параметры у тэга `<BODY>`, можно изменять внешний вид всей web-странички.

Тэг `<BODY>` может иметь следующие параметры:

- **bgcolor** – задает цвет фона страницы;
- **text** – задает цвет текста всей страницы по умолчанию;
- **link** – задает цвет ссылки, еще не посещенной;
- **vlink** – задает цвет ссылки, уже посещенной;
- **alink** – задает цвет активной ссылки (в момент нажатия на нее);
- **background** – задает фоновый рисунок страницы;
- **leftmargin**, **rightmargin**, **topmargin**, **bottommargin** – задают левое, правое, верхнее и нижнее поля страницы соответственно (в пикселях, px).

Рассмотрим параметры более подробно.

Задание цвета фона и текста странички

Цвет можно задавать тремя способами:

1. Через английский эквивалент названия цвета. Например,

`<BODY bgcolor="red">`

задает красный цвет фона.

2. Через RGB-формат `"#XXXXXX"`, где X - это число от 0 до F (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F), первая пара цифр отвечает за насыщенность красного цвета, вторая пара – за насыщенность зеленого, третья пара – синего. Например,

`<BODY bgcolor="#FF0000">`

задает красный цвет фона.

3. Через RGB-формат `rgb(X,X,X)`, где X – это число от 0 до 255, первая цифра отвечает за насыщенность красного цвета, вторая – за насыщенность зеленого, третья – синего. Например,

`<BODY bgcolor=rgb(255,0,0)>`

задает красный цвет фона.

Примечание. Основные оттенки цветов web-палитры можно посмотреть в Приложении I.

Упражнение 2.1

Создайте HTML-документ *format.html*. В нем задайте желтовато-коричневый цвет фона и темно-зеленый цвет текста.

Для этого:

- В конце пособия откройте web-палитру цветов.
- Найдите в ней соответствующие значения нужных цветов: желтовато-коричневый – #D2B48C, темно-зеленый – darkgreen.
- Для задания цвета фона и текста используйте параметры тэга <BODY> - **bbgcolor** и **text**:

```
<BODY bgcolor="#D2B48C" text="darkgreen">
```

- В теле странички напишите текст: «В языке HTML значения параметров тэгов, как правило, пишутся в кавычках. Но если в значении параметра нет пробелов, то кавычки можно и опустить. Нет правил без исключений!».

- Просмотрите страничку в окне браузера.

Задание фонового рисунка

Вы также можете повысить интерес к своей страничке, сделав фон страницы не одним цветом, а текстурой. Текстура - это небольшое изображение, которым «мостят» фон, т. е. оно многократно копируется как по вертикали, так и по горизонтали. Для создания текстур обычно применяют специальные графические редакторы, в которых есть инструменты создания текстур - текстуризаторы. Однако можно создать текстуру с помощью простых редакторов и инструментов, соблюдая следующие правила:

1. Рисунок не должен быть очень контрастным, т. е. пестрым.
2. Рисунок либо не должен обрываться на границе, либо его границы должны «сшиваться».
3. Если рисунок яркий, то шрифт документа должен быть темным, и наоборот.
4. Размер «плитки» текстуры выбирайте в пределах от 3х3 до 5х5 см.

Текстурный рисунок должен быть сохранен в формате GIF или JPG, после чего в параметре **background** тэга <BODY> нужно указать путь к фоновому рисунку. Например,

```
<BODY background="fon.gif">
```

Примечание. Данный пример говорит о том, что картинка лежит в том же каталоге (директории, папке), в которой лежит и наш документ.

Если картинка лежит в поддиректории (например, "images"), то путь к ней будет выглядеть так:

```
<BODY background="images/fon.gif">
```

Если картинка лежит на уровень выше, а документ находится в поддиректории, то путь к ней будет выглядеть так:

```
<BODY background="../fon.gif">
```

Если картинка лежит на другом сайте, то путь прописывается полностью:


```
<BODY background="http://www.mypage/images/fon.gif">
```

Для вашего удобства размещайте картинку в той же директории, что и документ, тогда путаницы будет меньше.

Упражнение 2.2

Страничке *format.html*, созданной в упр. 2.1, задайте фоновый рисунок (рис. 2.1).

Для этого:

- В графическом редакторе «Paint» создайте простейший фоновый рисунок. (Например, .

- Сохраните его в ту папку, где находится ваша страничка под именем *fon.jpg*.

- В коде странички у тэга `<BODY>` добавьте параметр `background`:

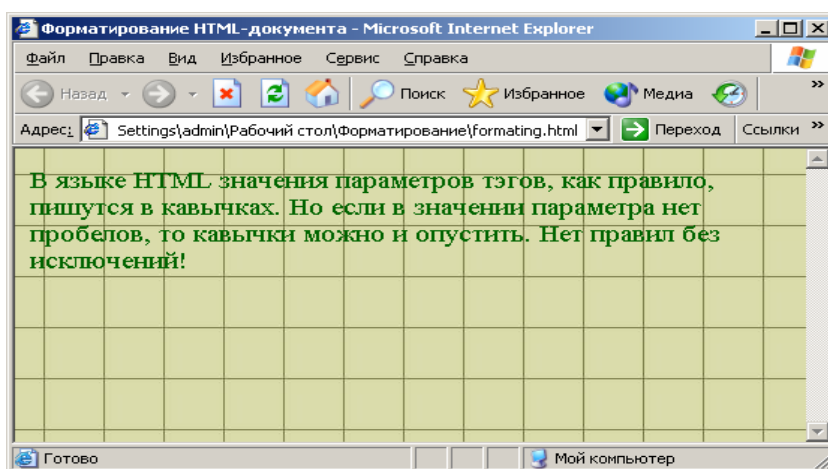


Рис. 2.1. HTML-документ с заданным фоновым рисунком

```
<BODY bgcolor="#DBDCAB" background="fon.jpg" text="darkgreen">
```

- Не забудьте про заголовок странички: «Форматирование HTML-документа».
- Посмотрите результат в окне браузера.

Задание полей странички

Поля страницы (отступы текста от границ окна браузера) задаются через параметры **leftmargin**, **rightmargin**, **topmargin**, **bottommargin**. Значения этих параметров указываются в пикселях. Например,

```
<BODY leftmargin="100" topmargin="40">
```

Упражнение 2.3

У созданного в предыдущих упражнениях HTML-документа *format.html* задайте поля: левое – 100 пикселей, правое – 80 пикселей, верхнее и нижнее – 50 пикселей.

Для этого:

- У тэга `<BODY>` опишите новые параметры (выделено полужирным шрифтом):

```
<BODY    bgcolor="#DBDCAB"    text="darkgreen"    leftmargin="100"
rightmargin="80" topmargin="50">
```

- Посмотрите результат в окне браузера.

Задание 2.1

Создайте HTML-документ *pautina.html* с названием «Всемирная паутина: первое знакомство». Отформатируйте документ: цвет фона - голубовато-стальной, цвет текста - блекло-голубой, поля страницы – все по 50 пикселей. В тело странички внесите текст:

«WWW (World Wide Web) – территориально распределенная гипертекстовая система Internet, или «Всемирная паутина». В сети Internet это сервис, предназначенный для доступа к информации, организованной при помощи гипертекстовых связей».

Задание 2.2

Задайте фоновый рисунок у HTML-документа *pautina.html*.

Форматирование абзаца

Параграфы

Предположим, что вы хотите разбить текст на параграфы. Для этого нужно перед началом параграфа поставить тэг `<P>`. Когда браузер обнаружит этот тэг, то он сам вставит перед началом параграфа пустую строку. Тэг параграфа – парный, и у открывающего тэга `<P>` существует закрывающий `</P>`.

Тэг `<P>` имеет следующие параметры:

- **title** – краткое описание параграфа (всплывает в виде подсказки при наведении на текст параграфа);
- **align** – задает выравнивание абзаца. Параметр **align** может принимать следующие значения:
 - *center* - текст абзаца выравнивается по центру;
 - *left* - по левому краю;
 - *right* - по правому краю;
 - *justify* - по ширине экрана.

Упражнение 2.4

Создайте HTML-документ *paragraph.html* с названием странички «Параграфы». В этот документ добавьте текст, используя разбиение на абзацы с различным выравниваем:

<P align=left title="Выравнивание по левому краю">Этот текст выровнен по левому краю, при этом правый край окажется неровным, «рваным». Такой текст хотя и выглядит не очень аккуратным, но его все-таки легко читать.**</P>**

<P align=right title="Выравнивание по правому краю"> Этот абзац выровнен по правому краю. Такое выравнивание текста значительно затрудняет чтение, но привлекает к себе внимание. Поэтому его часто используют для оформления заголовков и эпиграфов.**</P>**

<P align=center title="Выравнивание по центру">А текст, выровненный по центру, используют для оформления коротких заголовков, но большой текст в таком виде читать очень тяжело.**</P>**

Просмотрите ее в окне браузера.

Примечание. Обратите внимание на то, что значение у параметра *align* указано без кавычек, а у параметра *title* значение задано в кавычках. Почему? Подсказка: прочтите текст странички *format.html*, созданной в упр. 2.1.

Заголовок

Большинство документов имеют заголовок. Для его создания используют тэги **<Hx></Hx>**, где *x* – число от 1 до 6. Заключив текст между этими тэгами, вы получите заголовок соответствующего уровня.

Упражнение 2.5

Создайте HTML-документ *headings.html* и добавьте в него примеры встроенных заголовков различного уровня:

```
<H1>Заголовок 1</H1>
<H2>Заголовок 2</H2>
<H3>Заголовок 3</H3>
<H4>Заголовок 4</H4>
<H5>Заголовок 5</H5>
<H6>Заголовок 6</H6>
```

Просмотрите результат в окне браузера (см. рис. 2.2).

Задание 2.3

В HTML-документ *pautina.html*, созданный в зад. 2.1, добавьте заглавие текста «Всемирная паутина: первое знакомство». Отформатируйте заглавие с помощью встроенных заголовков.

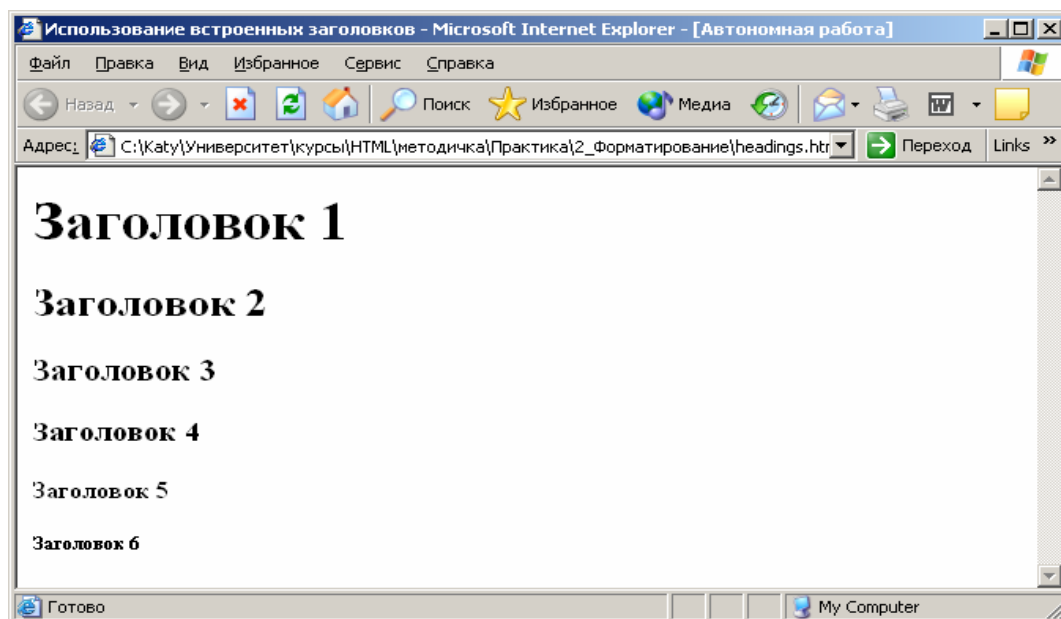


Рис. 2.2. Встроенные заголовки HTML-документа

Принудительный перенос на новую строку

Браузеры игнорируют множественные пробелы и символы конца строки. Для принудительного переноса на следующую строку используется тэг **
**. С помощью этого тэга также можно вставлять пустые строки.

Упражнение 2.6

Создайте HTML-документ *romans3.html*, содержащий текст стихотворения Игоря Северянина «Романс 3».

Для этого:

- Каждое четверостишие начинайте новым абзацем (тэг **<P>**).
- Используйте тэг **
** для принудительного переноса строк внутри четверостиший.
- Название стихотворения оформите встроенным заголовком 3-го уровня.
- Подпись автора стихотворения выровняйте по правому краю.

```
<H3>РОМАНС III</H3>
```

```
<P>За каждую строку, написанную кровью,<BR>
```

```
За каждую улыбку обо мне, -<BR>
```

```
Тебе отвечаю спокойною любовью<BR>
```

```
И образ твой храню в душевной глубине.</P>
```

```
<P>Не видимся ли миг, не видимся ль столетье –<BR>
```

```
Не все ли мне равно, не все ль равно тебе,<BR>
```

```
Раз примагничены к бессмертью цветоплетью<BR>
```

```
Сердца углубные в медузовой алчбе?..</P>
```

```
<P>О, да: нам все равно, что мы с тобой в разлуке,<BR>
```

```
Что у тебя есть муж, а у меня – жена.<BR>
```

```
Ищи забвения в искусстве и в науке.<BR>
```

```
И в сновидениях, и в грезности вина.</P>
```

```
<P align=right>Игорь Северянин</P>
```

- Посмотрите страничку в окне браузера.

Текст с отступом

Текст, заключенный между тэгами `<BLOCKQUOTE>` и `</BLOCKQUOTE>`, выводится браузером на экран с увеличенным левым полем.

Упражнение 2.7

Создайте HTML-документ *viveskam.html* со стихотворением Владимира Маяковского «Вывескам»:

Вывескам

Читайте железные книги!
Под флейту золоченой буквы
Полезут копченые сиги
И золотокудрые брюквы.
 А если веселостью песьей
 Закружат созвездия «Магги» -
 Бюро похоронных процессий
 Свои проведут саркофаги.
Когда же, хмур и плачевен,
Загасил фонарные знаки,
Влюбляйтесь под небом харчевен
В фаянсовых чайников маки!

Для этого:

- Отформатируйте название стихотворения встроенным заголовком, например, 4-го уровня:

```
<H4>Вывескам</H4>
```

- Четверостишия стихотворения оформите с использованием тэга `
` (как в упр. 2.6).

- Для отступа второго четверостишия заключите его текст между тэгами `<BLOCKQUOTE>` и `</BLOCKQUOTE>`:

```
<BLOCKQUOTE>А если веселостью песьей<BR>  
Закружат созвездия “Магги” - <BR>  
Бюро похоронных процессий<BR>  
Свои проведут саркофаги.</BLOCKQUOTE>
```

- Посмотрите результат в окне браузера.

Предварительно форматированный текст

Как уже было сказано ранее, браузеры игнорируют множественные пробелы и символы конца строки. Из этого правила, однако, есть исключение. Текст, заключенный между метками `<PRE>` и `</PRE>` (от англ. *preformatted* - предварительно форматированный), выводится браузером на экран как есть - со всеми пробелами, символами табуляции и конца строки.

Упражнение 2.8

Добавьте на вашу страничку стихотворение Маяковского, сохраняя все отступы и переносы, для сохранения отступов и переносов используйте тэг `<PRE>`:

`<PRE>`

Только

солнце усядется,

канув

за опустевшие

фабричные стройки,

стонут

окраины

от хулиганов

вроде вот этой

милой тройки.

Владимир Маяковский

`</PRE>`

Посмотрите результат в окне браузера.

Примечание. Обратите внимание, что в предварительно форматированном тексте по умолчанию используется шрифт фиксированной ширины (*Courier*).

Форматирование символов

Форматирование символов можно условно разделить на логическое и физическое. При **логическом форматировании** разработчик web-странички структурирует документ, выделяет логические части, одновременно изменяя внешний вид символов. При **физическом форматировании** разработчик не учитывает контекста и лишь придает некоторым символам конкретный вид.

Применяя тэги логического или физического форматирования, мы в принципе получаем одинаковый результат. Но логическое форматирование и четкое структурирование являются признаками хорошо составленного документа. При использовании логических тэгов в ближайшем будущем станет возможным, например, автоматический поиск цитат в сети Internet.

Логическое форматирование

Для логического форматирования существует группа формирующих тэгов.

`<CITE>` - используется для выделения цитат или названий книг и статей, при этом текст обычно выводится курсивом. Например:

Я прочитал `<CITE>`Войну и мир`</CITE>`

<CODE> - применяется для вывода небольшого куса программного кода шрифтом фиксированной длины (для больших листингов используется тэг **<PRE>**). Например,

Одну из первых строчек, которые узнает программист, это:
<CODE>Print "Hello, World!"**</CODE>**

**** - используется для выделения важных фрагментов текста. Браузер обычно отображает такой текст *курсивом*. Например,

Изменение внешнего вида символов и абзацев называется
****форматированием****

<KBD> - выделяет шрифтом фиксированной ширины текст, который предлагают ввести пользователю с клавиатуры. Например,

Для запуска программы-кодировщика введите в поле слово
<KBD>Кодировка**</KBD>**

<SAMP> - используется для выделения нескольких символов шрифтом фиксированной ширины. Например,

<SAMP>ABC**</SAMP>** - этими буквами в Америке обозначают алфавит.

**** - используется для выделения важных фрагментов текста **полужирным шрифтом**.

<VAR><VAR> - используется для отметки имен переменных. Обычно такой текст выделяется *курсивом*. Например,

Пусть задана функция: **<VAR>**y=2x**<VAR>**

<ADDRESS></ADDRESS> - используется при выделении почтового адреса *курсивом*.

<STRIKE></STRIKE> - используется для **зачеркивания** текста.

Упражнение 2.9

Создайте HTML-документ *logical.html*. Внесите в него тексты приведенных выше примеров и просмотрите страничку в окне браузера.

Физическое форматирование

Для физического форматирования используются следующие тэги:

- **** - **полужирный** шрифт
- **<I></I>** - *курсив*
- **<U></U>** - подчеркнутый
- **<S></S>** - выделение **неречеркиванием**

- `<TT></TT>` - текст отображается шрифтом фиксированной ширины (чаще всего шрифт Courier)
- `<BIG></BIG>` - текст отображается БОЛЬШЕ, чем основной шрифт
- `<SMALL></SMALL>` - текст отображается меньше, чем основной шрифт
- `` - нижний индекс
- `` - верхний индекс

Упражнение 2.10

Создайте HTML-документ *physical.html* и добавьте на страничку текст с использованием описанных выше тэгов физического форматирования:

`<P>`При использовании тэгов начертания текст может быть ``полужирным``, `<I>`курсивным`</I>` и `<U>`подчеркнутым`</U>`. Также можно сделать `<I>`полужирный курсив`</I>`, `<U>`полужирный подчеркнутый`</U>` и даже `<I><U>`полужирный подчеркнутый курсив!`</U></I>`. Кроме того, текст можно `<S>`зачеркнуть.`</S>`!`</P>`

Просмотрите результат в окне браузера.

Примечание. Обратите внимание, что при использовании комбинации парных тэгов следует быть внимательней с порядком их расположения. Последовательность закрывающих тэгов должна быть «зеркальной» по отношению к последовательности открывающих тэгов. То есть, если первым из открывающих тэгов стоит ``, а затем `<I>`, то сначала следует написать тэг `</I>`, а затем уже ``.

Упражнение 2.11

Создайте HTML-документ *formula.html* и добавьте в него химическую формулу воды – H_2O и математическую формулу кубической функции – $y=x^3$ (рис. 2.3).

- Для создания верхнего и нижнего индексов используйте соответствующие тэги `<SUP>` и `<SUB>`.

- Результат просмотрите в окне браузера.

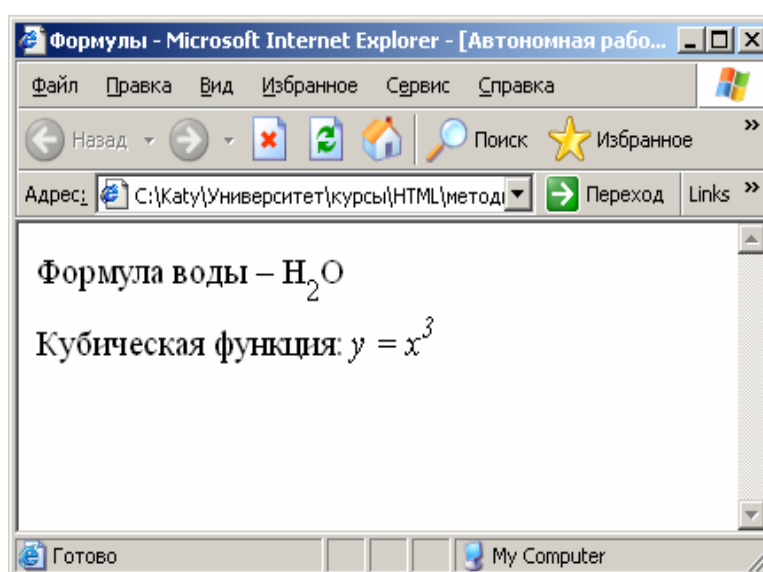


Рис. 2.3. Использование тэгов нижнего и верхнего индексов

Работа со шрифтом

Некоторые тэги вообще не имеет смысла применять без атрибутов. К таким тэгам можно отнести тэг ****. Тэг **** применяется при работе со шрифтом и имеет следующие параметры:

- **size** – задает размер шрифта (задается в пунктах; по умолчанию размер шрифта принят равным 3);
- **face** – задает тип шрифта. Очень аккуратно используйте этот атрибут, так как заданный шрифт должен присутствовать на компьютере пользователя - в противном случае браузер подставит шрифт, определенный по умолчанию (как правило, это Times New Roman). Применяйте шрифты, в наличии которых вы уверены, иначе пользователь увидит текст иначе, чем вы. К стандартным шрифтам можно отнести шрифты, поставляемые с Windows 95/98, MS Office;
- **color** – задает цвет шрифта. Способы задания цвета были рассмотрены в разделе «Настройка внешнего вида страницы».

Упражнение 2.12

Работа с размером шрифта. Создайте HTML-документ *font.html* и добавьте текст в тело странички:

```
<P><FONT size="1">Увеличиваем</FONT>  
<FONT size="3">размер</FONT>  
<FONT size="5">символов</FONT>  
<FONT size="6">текста</FONT></P>
```

Просмотрите страничку в окне браузера.

Упражнение 2.13

Работа с цветом шрифта. Добавьте текст в тело странички *font.html* и просмотрите ее в окне браузера:

```
<P><FONT color="red">красный</FONT> - стой!</P>  
<P><FONT color="yellow">желтый</FONT> - внимание!</P>  
<P><FONT color="green">зеленый</FONT> - иди!</P>
```

Задание 2.4

Создайте HTML-документ *rainbow.html* и внесите в него текст:

Каждый **ОХОТНИК** **ЖЕЛАЕТ** **ЗНАТЬ**, где сидит фазан.

Каждое слово окрасьте в соответствующий цвет радуги.

Специальные символы

Некоторые символы невозможно просто набрать с клавиатуры: это буквы алфавитов части европейских языков, математические и некоторые другие символы. Некоторые символы, введенные в HTML-документ, могут быть неверно поняты браузером: например, символы «<» и «>» используются браузером для указания тэгов и, возможно, могут быть интерпретированы только так.

Вместо символов этой группы необходимо использовать замену. Например,

Символ	Замена
<	<
>	>

Примечание. Более полную таблицу специальных символов можно посмотреть в приложении к методическому пособию.

Упражнение 2.14

Создайте HTML-документ *symbols.html* с заголовком «Использование замены спецсимволов».

В тело документа внесите текст, используя замены символам < и >:

Тэг **
** используется для принудительного переноса строк. Текст, заключенный между парными тэгами **<BLOCKQUOTE>** и **</BLOCKQUOTE>**, будет иметь отступ от левого края окна браузера.

Посмотрите результат в окне браузера.

Задание 2.5

Создайте HTML-документ *functions.html*, содержащий следующие формулы (для специальных символов <, >, α , β , \neq используйте замену):

Функции сокращенного умножения:

$$a^2 - b^2 = (a+b)(a-b)$$

$$(a+b)^2 = a^2 + 2ab + b^2$$

Логарифмы:

Запись $\log_a b = x$ означает, что $a^x = b$; здесь $a > 0$, $b > 0$, $a \neq 2$

Тригонометрия:

$$\sin(\alpha + \beta) = \sin \alpha * \cos \beta + \cos \alpha * \sin \beta$$

Создание бегущей строки

Текст абзаца можно сделать «движущимся», т. е. оформить его в виде бегущей строки. Для этого используйте тэг **<MARQUEE>..</MARQUEE>**

Тэг **<MARQUEE>** имеет следующие параметры:

- **bgcolor** – задает цвет фона;

- **height** – задает высоту фоновой полосы (в пикселях или процентах от высоты окна браузера);
- **width** – ширина полосы бегущей строки (в пикселях или процентах от ширины окна браузера);
- **direction** – задает направление бегущей строки:
 - *left* – справа налево (это значение используется по умолчанию),
 - *right* – слева направо,
 - *up* – снизу вверх,
 - *down* – сверху вниз.
- **behavior** – управляет поведением бегущей строки:
 - *scroll* – дойдя до края, строка уходит из поля зрения, затем появляется с противоположной стороны,
 - *slide* – строка, достигнув края окна, останавливается,
 - *alternate* – строка, достигнув противоположного края окна, меняет свое направление.
- **hspace, vspace** – задает горизонтальный и вертикальный соответственно отступы у бегущей строки (в пикселях);
- **loop** – задает количество переходов строки по экрану;
- **scrollamount** – задает скорость движения строки (в пикселях в секунду);
- **scrolldelay** – определяет временной интервал (в миллисекундах) между шагами.

Упражнение 2.14

Добавьте в начало вашей странички бегущую строку «Всем привет! Меня зовут...».

- Для этого заключите текст строки между тэгами `<MARQUEE>` и `</MARQUEE>`

```
<P><MARQUEE>Всем привет! Меня зовут...</MARQUEE></P>
```

- Отформатируйте строку: измените размер шрифта, начертание, цвет текста.
- Изменяя значения атрибутов тэга `<MARQUEE>`, измените поведение бегущей строки: при достижении границы окна она должна менять направление, увеличьте скорость перемещения строки.

Задание 2.6

Создайте страничку *hello.html* с бегущей строкой «Привет!», которая двигается в различных направлениях, подобно бильярдному шарiku, отталкиваясь от края окна браузера и меняя свое направление.

Примечание. Используйте вложенную конструкцию `<MARQUEE>` с различными направлениями.

Списки

Очень важный элемент web-страничек – это списки. Они бывают: **нумерованными**, например:

- Повышение квалификации:
1. Кадровик.
 2. Секретарь.
 3. Заведующий магазинами.
 4. Маркетолог.

маркированными, например:

- Предлагаем стоматологические услуги:
- протезирование зубов;
 - удаление зубов без боли;
 - рентген;
 - детский прием.

смешанными, т. е. сочетать в себе элементы того и другого списка.

В **нумерованном** списке (*Ordered List*) каждому элементу предшествует его порядковый номер. Для организации нумерованного списка используются следующие тэги:

**** - открывает нумерованный список

**** - задает элемент списка (*List Item*)

**** - закрывает список

Тэг **** имеет два параметра:

- **start** – задает число, с которого начинается нумерация (по умолчанию с 1);
- **type** – задает тип нумерации (по умолчанию арабская нумерация 1,2,3,...):
 - type=**“i”** – нумерация римскими малыми (i, ii, iii, ...);
 - type=**“I”** – нумерация римскими заглавными (I,II,III,...);
 - type=**“a”** – нумерация английскими малыми (a, b, c, ...);
 - type=**“A”** – нумерация английскими заглавными (A, B, C,...).

В **маркированном** списке (*Unordered List*) каждому элементу списка предшествует маркер. Для создания маркированного списка используют следующие тэги:

**** - открывает список

**** - задает элемент списка

**** - закрывает список

Тэг **** тоже имеет параметр **type**, с помощью которого можно задавать вид маркера:

- type=**“circle”** – в списке используется маркер °;
- type=**“square”** – в списке используется маркер ■;
- type=**“disc”** – в списке используется маркер • (этот маркер используется по умолчанию).

Упражнение 2.15

Создайте HTML-документ *orderedlist.html* и добавьте в него нумерованный список, указанный в примере:

```
<P>Повышение квалификации </P>
<OL>
<LI>Кадровик.</LI>
<LI>Секретарь.</LI>
<LI>Заведующий магазинами.</LI>
<LI>Маркетолог.</LI>
</OL>
```

- Просмотрите результат в окне браузера.
- С помощью параметра `type` задайте нумерацию римскими заглавными:

```
<OL type="I">
```

- Просмотрите результат в окне браузера: нумерация сменилась на заглавные римские цифры.

Упражнение 2.16

Создайте HTML-документ *unorderedlist.html* и добавьте в него маркированный список, указанный в примере:

```
<P>Предлагаем стоматологические услуги:</P>
<UL>
<LI>протезирование зубов;</LI>
<LI>удаление зубов без боли;</LI>
<LI>рентген;</LI>
<LI>детский прием.</LI>
</UL>
```

- Просмотрите результат в окне браузера.
- С помощью параметра `type` измените стандартный вид маркера на квадратик ■:

```
<UL type="square">
```

- Просмотрите результат в окне браузера: маркер с кружка «●» сменился на квадратик «■».

Упражнение 2.17

Создайте HTML-документ *kurs.html* и добавьте в него смешанный список: во внешнем нумерованном списке используйте нумерацию заглавными римскими цифрами, во вложенном маркированном – маркер «диск» - ● (см. рис. 2.12):

```
<P>Изучение курса HTML включает в себя:</P>
<OL type="I">
<LI>Введение в HTML</LI>
<LI>Структура HTML-документа</LI>
<LI>Форматирование HTML-документа</LI>
```



```
<UL type="disc">
  <LI>Настройка внешнего вида страницы</LI>
  <LI>Форматирование абзацев</LI>
  <LI>Форматирование символов</LI>
  <LI>Списки</LI>
</UL>
<LI>Вставка рисунков в HTML-документа</LI>
<LI>Работа с таблицами</LI>
<LI>Ссылки</LI>
<LI>Фреймы</LI>
<LI>Формы</LI>
<LI>Каскадные таблицы стилей (CSS) </LI>
<LI>Классы</LI>
<LI>Слои и позиционирование</LI>
</OL>
```

Просмотрите результат в окне браузера (рис. 2.4).

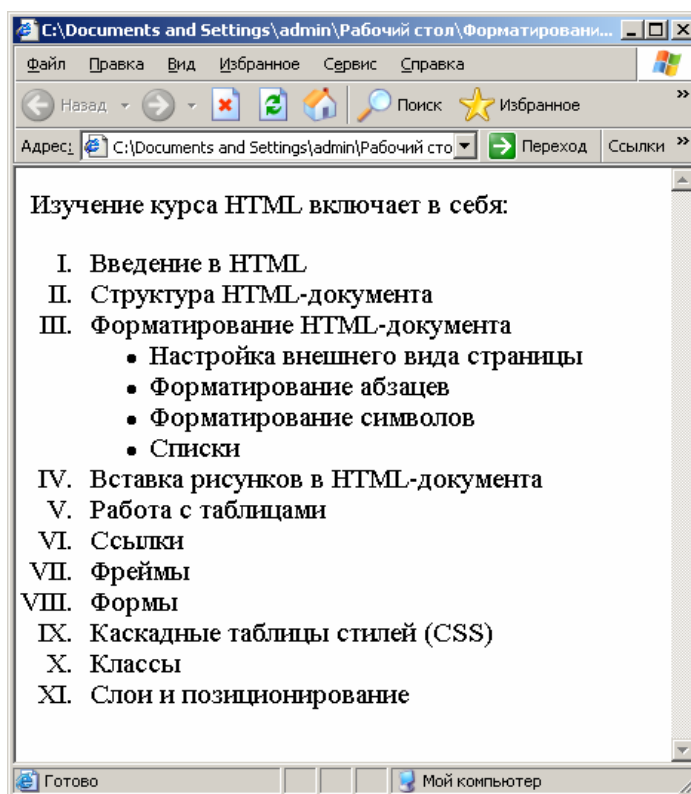


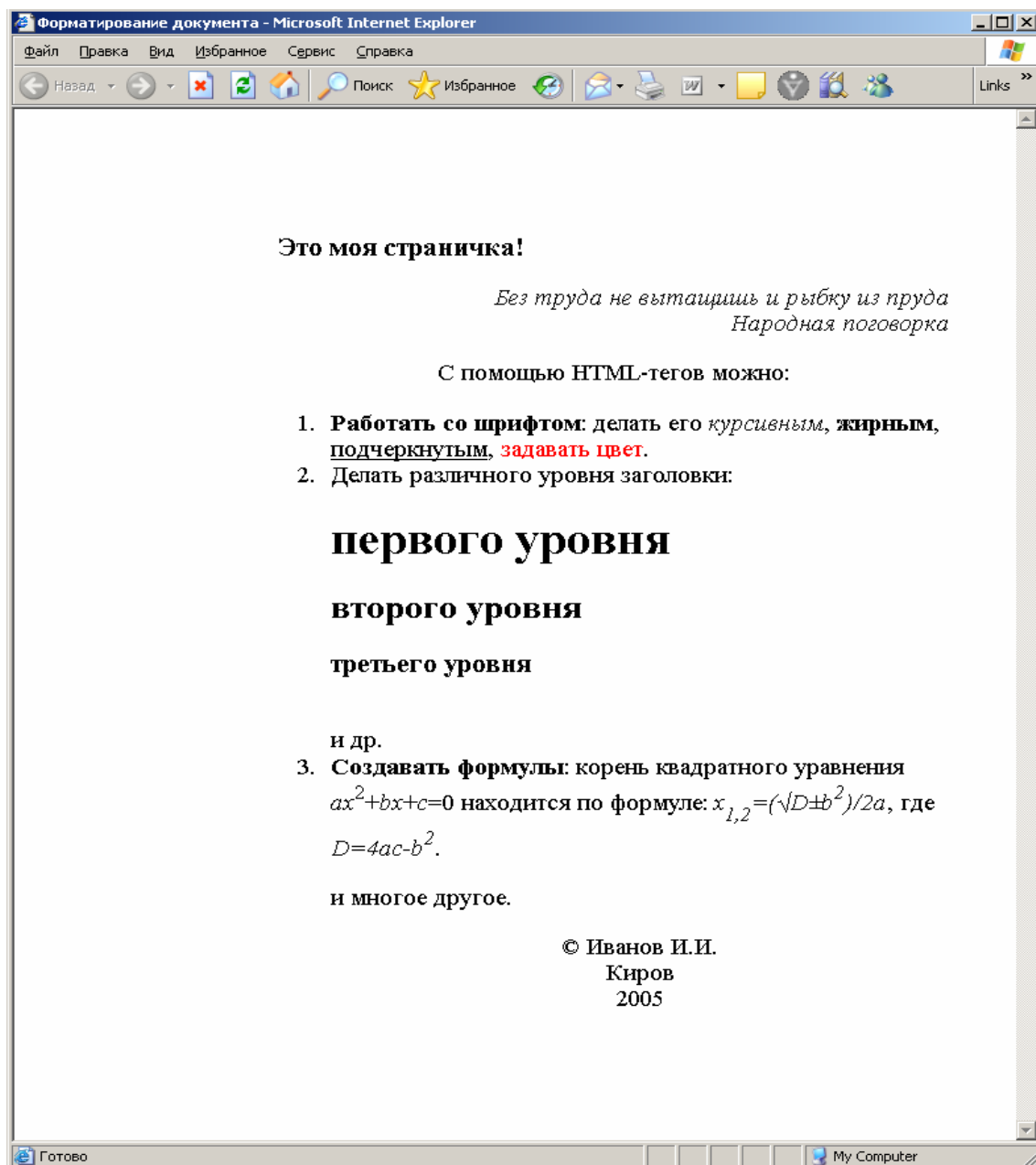
Рис. 2.4. Смешанный список в HTML

Задание 2.7

Создайте HTML-документ *about.html* и перечислите ваши увлечения, используя маркированный список, и дисциплины, которые вы изучаете в университете, используя нумерованный список. Отформатируйте текст: задайте цвет, тип шрифта, начертание, размер шрифта. В конце документа добавьте строчку:

© Иванов И.И. 2005г

Для вставки специального символа © используйте замену на соответствующий код или имя.

Рис. 2.6. Образец HTML-документа *formatting.html*

Глава 3. Графика в HTML-документе

Использование рисунков

Большинство браузеров могут вместе с текстом показывать рисунки форматов gif (*.gif), jpg, jpeg (*.jpg), png (*.png) или bmp (*.bmp) .

Поскольку каждый рисунок занимает много времени на отображение на экране (что замедляет загрузку всей странички), не рекомендуется включать слишком большое количество или чрезмерно большие по размеру рисунки в ваш HTML-документ.

Вставка рисунков

Для вставки картинки используется непарный тэг .

src – обязательный параметр тэга , в котором указывается путь к картинке. Если картинка находится в той же папке, что и web-страничка, то достаточно написать имя файла.

Примечание. *Подробнее о формировании пути к рисунку смотрите в предыдущей главе в пункте «Задание фонового рисунка».*

Упражнение 3.1

Создайте новый HTML-документ *picture.html* с названием странички «Вставка рисунков в HTML-документ» и разместите на нем рисунок.

Для этого:

- Добавьте в тело странички абзац с текстом, комментирующим, что изображено на рисунке. Например:

<P>Перед вами изображена цапля</P>

- Затем, используя тэг , добавьте на страничку рисунок:

- Посмотрите результат в окне браузера (рис. 3.1)

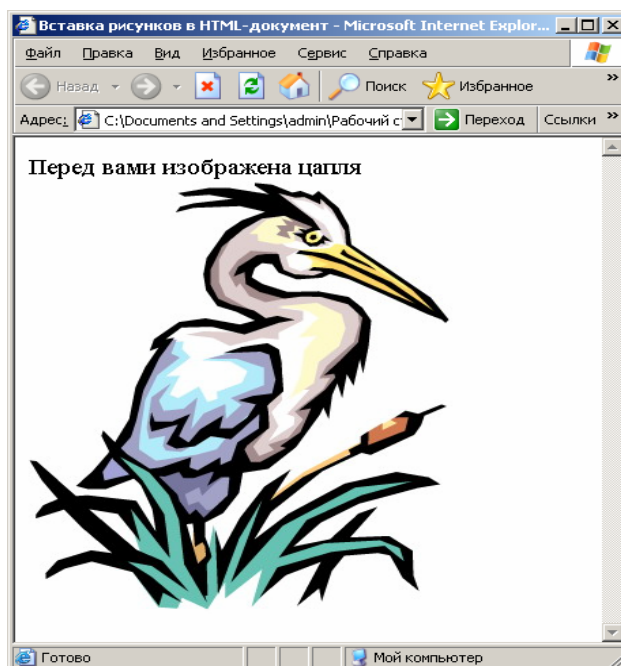


Рис. 3.1. Рисунок, добавленный в HTML-документ

Дополнительные параметры графических изображений

Тэг имеет дополнительные параметры, с использованием которых можно форматировать рисунок:

- **width** – задает ширину рисунка;
- **height** – задает высоту рисунка;
- **align** – задает положение рисунка относительно абзаца, внутри которого помещен рисунок. Данный тэг может принимать значения:
 - **top** – вывод рисунка на уровне самого высокого элемента абзаца;
 - **middle** – вывод рисунка по центру строки абзаца;
 - **center** – вывод рисунка по центру строки абзаца;
 - **bottom** – вывод рисунка на уровне самого низкого элемента абзаца;
 - **left** – вывод рисунка слева от абзаца;
 - **right** – вывод рисунка справа от абзаца.
- **hspace**, **vspace** – задает горизонтальный и вертикальный отступы текста от изображения (задается в пикселях);
- **border** – задает границу картинки (измеряется в пикселях, по умолчанию border=1, отсутствие границы – border=0);
- **alt** – задает альтернативный текст картинки.

Упражнение 3.2

Уменьшите размеры рисунка, который вы поместили в HTML-документе *picture.html* в упр. 3.1.

Для этого:

- У тэга добавьте параметры width и height с нужными значениями. Например,

```
<IMG src="bird.jpg" width=50 height=100>
```

- Будьте осторожны – при изменении размеров соблюдайте пропорции!

Преимущества и недостатки уменьшения размеров изображений средствами HTML

В уменьшении размеров рисунка средствами HTML есть плюсы и минусы.

Минусы уменьшения размеров рисунка при помощи HTML:

- рисунок сохраняет свои реальные размеры и, соответственно, его «вес» не изменится, даже если вы сделаете огромный рисунок, который «весит» 1 Мб, размером с почтовую марку. А следовательно и вес странички будет большим;
- картинка потеряет много деталей, что в некоторых случаях превращает ее в нечто непонятное.

Плюсы уменьшения размеров рисунка при помощи HTML:

- рисунок можно вписать в дизайн, уменьшив его размер, а при копировании на локальный диск он сохранит свои размеры.

Совет будущему HTML-редактору. При грамотном размещении на странице рисунков лучше указать еще и их ширину и высоту. В этом случае браузеру не нужно будет выполнять работу по определению размера рисунка, что значительно ускорит загрузку страницы.

Упражнение 3.3

Работа с положением рисунка относительно текста абзаца.

В созданный в предыдущих упражнениях HTML-документ *picture.html* добавьте еще один абзац текста, а рисунок переместите внутрь этого абзаца (между тэгами `<P>` и `</P>`):

```
<P><IMG src=bird.jpg width=100 height=117>Цапля обитает на болотах,
имеет длинные ноги. Постоянно стоит на одной ноге, чтобы не спугнуть
свою добычу - лягушек.</P>
```

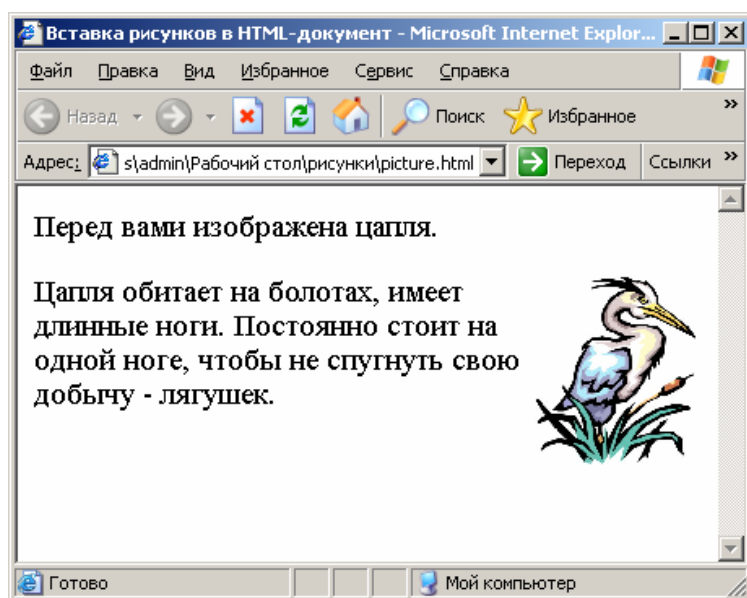


Рис. 3.2. Положение рисунка
справа от текста абзаца

Задайте у рисунка положение справа относительно текста абзаца.

Для этого:

- У тэга `` добавляем параметр `align`, которому присваиваем значение `right`:

```
<IMG src=bird.jpg align=right>
```

- Посмотрите результат в окне браузера (рис. 3.2)

Задание 3.1

Изменяя значение параметра `align` у тэга ``, поэкспериментируйте с положением рисунка на страничке.

Примечание. Не забывайте, что положением рисунка можно управлять, если рисунок помещен внутрь абзаца!

Упражнение 3.4

Задайте у рисунка горизонтальный отступ текста – 40 пикселей.

Для этого:

- У тэга `` добавьте еще один параметр `hspace`:

```
<IMG src=bird.jpg width=100 height=117 align=left hspace=40>
```

- Посмотрите результат в окне браузера.

Альтернативный текст

Пользователь, путешествуя по сети Internet, в целях уменьшения количества скачанной информации может отключить отображение рисунков (команда меню браузера Internet Explorer **Сервис/Свойства обозревателя**, вкладка **Дополнительно** -> убрать галочку **Отображать рисунки**).

Тогда страничка с рисунками будет выглядеть как на рис. 3.3.

Согласитесь, что не очень-то приятно увидеть на месте рисунка непонятное белое пятно.

Чтобы избежать таких накладок, используют альтернативный текст, который выводится на том месте, где должен находиться «пропавший» рисунок. Цель альтернативного текста — объяснить, что изображено на рисунке.

Для добавления к рисунку альтернативного текста нужно использовать дополнительный параметр **alt** тэга ****. Например,

```
<IMG src="bird.jpg" alt="Цапля">
```

Тогда страничка будет выглядеть как на рис. 3.4.

по сети Internet, в целях уменьшения

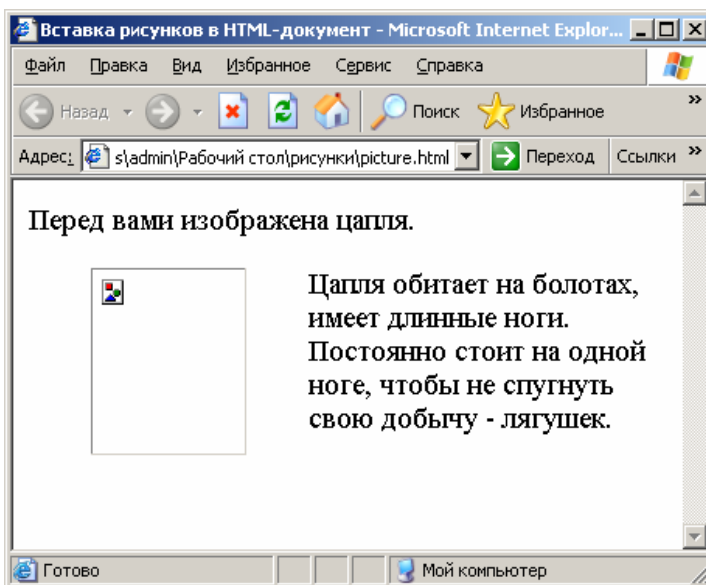


Рис. 3.3. HTML-документ при отключении режима отображения рисунков

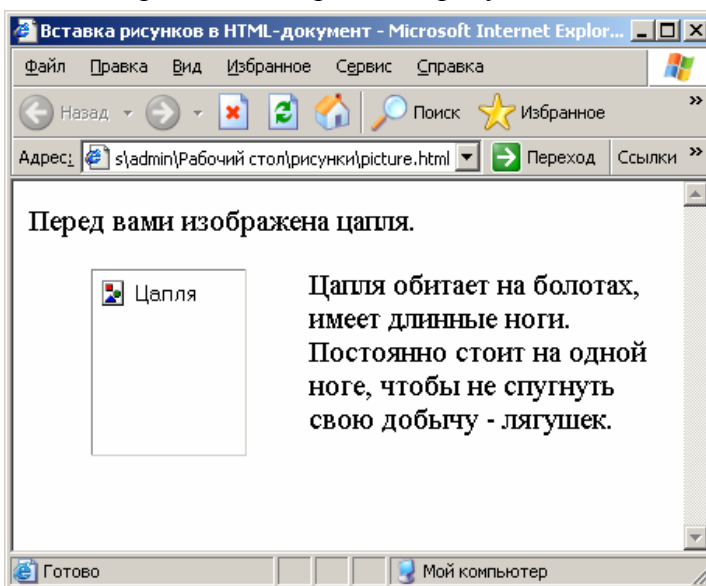


Рис. 3.4. Рисунок с альтернативным текстом

Упражнение 3.5

Придумайте для своего рисунка альтернативный текст.

Совет будущему HTML-редактору. Альтернативный текст полезен также тем, что он используется поисковыми системами при поиске картинок в сети Internet. Поэтому, если вы хотите, чтобы вашу страничку нашли даже по картинке, используйте альтернативный текст.

Тэг <HR>

С помощью тэга **<HR>** можно изобразить горизонтальную линию.

Тэг <HR> имеет ряд параметров, с помощью которых можно задать внешний вид линии:

- **width** – задает длину линии (измеряется в процентах от ширины окна браузера или пикселях);
- **size** – задает толщину линии (измеряется в пикселях);
- **align** – задает горизонтальное положение линии относительно окна браузера. Может принимать значения:
 - *right* – размещение относительного правого края окна браузера;
 - *left* – размещение относительно правого края окна браузера;
 - *center* – размещение по центру окна браузера.
- **color** – задает цвет линии.

Упражнение 3.6

В конце HTML-документа *picture.html* разместите красную линию толщиной 3 пикселя, длиной в 50% ширины окна браузера, размещенную по центру окна (рис. 3.5).

Для этого:

- Откройте код HTML-документа *picture.html*.
- В конце тела страницы напишите тэг <HR> с нужными параметрами:

```
<HR color=red size=3 width=50%
    align=center>
```

- Загрузите страницу в окне браузера и посмотрите результат.

Задание 3.2

Дополните оформление документа несколькими линиями различной толщины и длины (рис. 3.6)

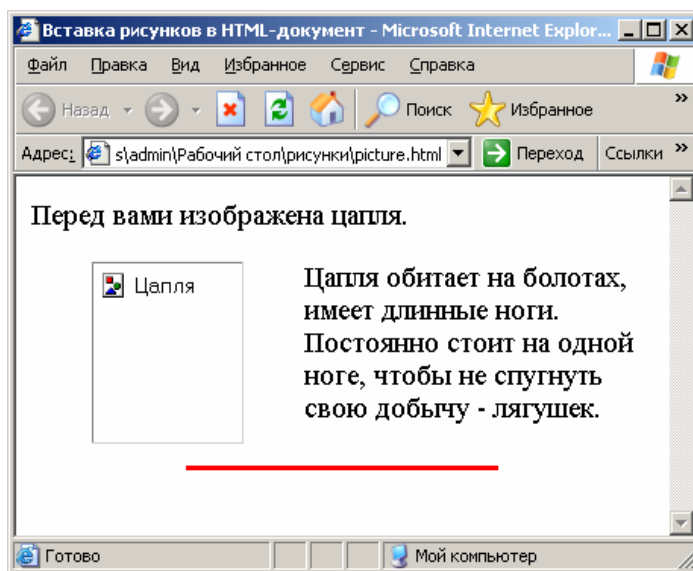


Рис. 3.5. Линия в HTML-документе

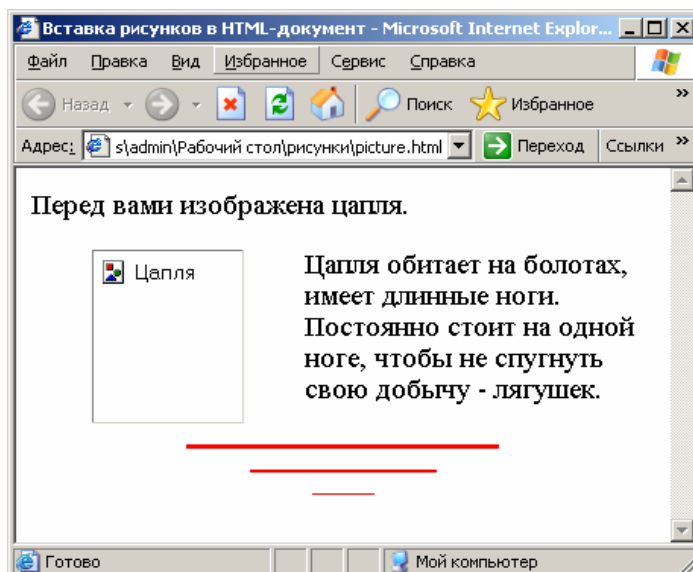


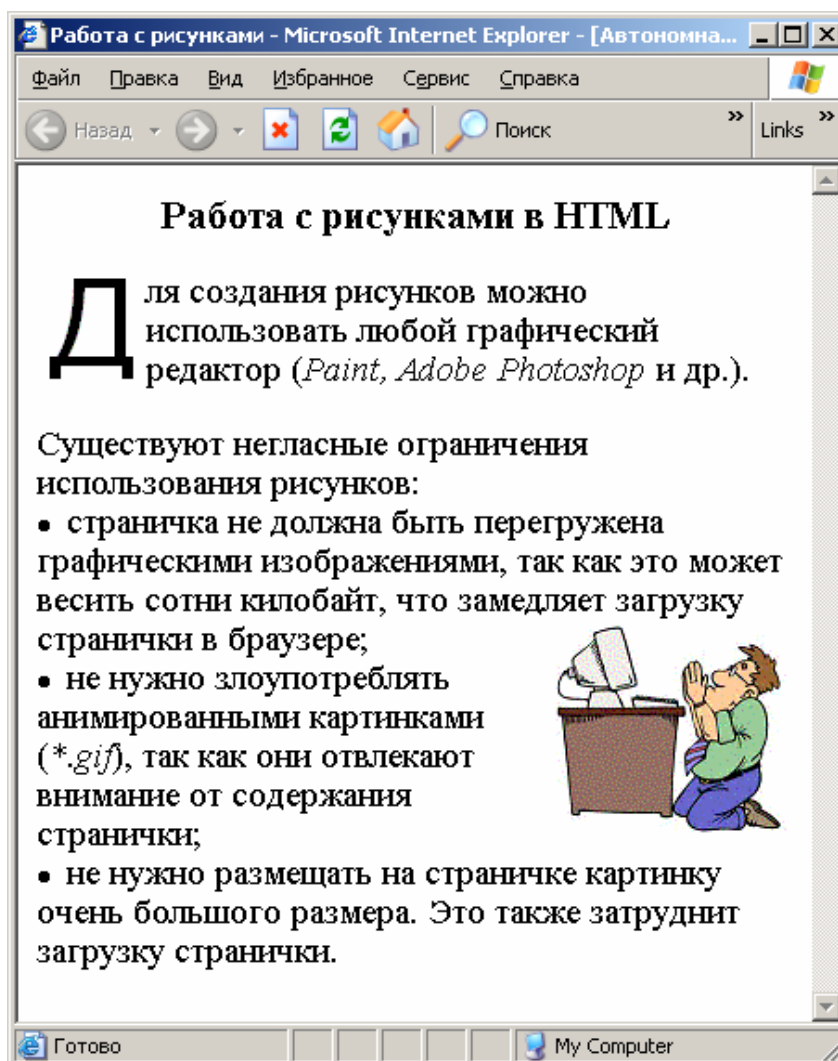
Рис. 3.6. Несколько линий в HTML-документе

Вопросы для повторения

1. Рисунки каких форматов способны отображать браузеры?
2. Какой обязательный параметр должен иметь тэг ? За что он отвечает?
3. Какие параметры относятся к необязательным?
4. Что следует учитывать при использовании рисунков на web-страничке?

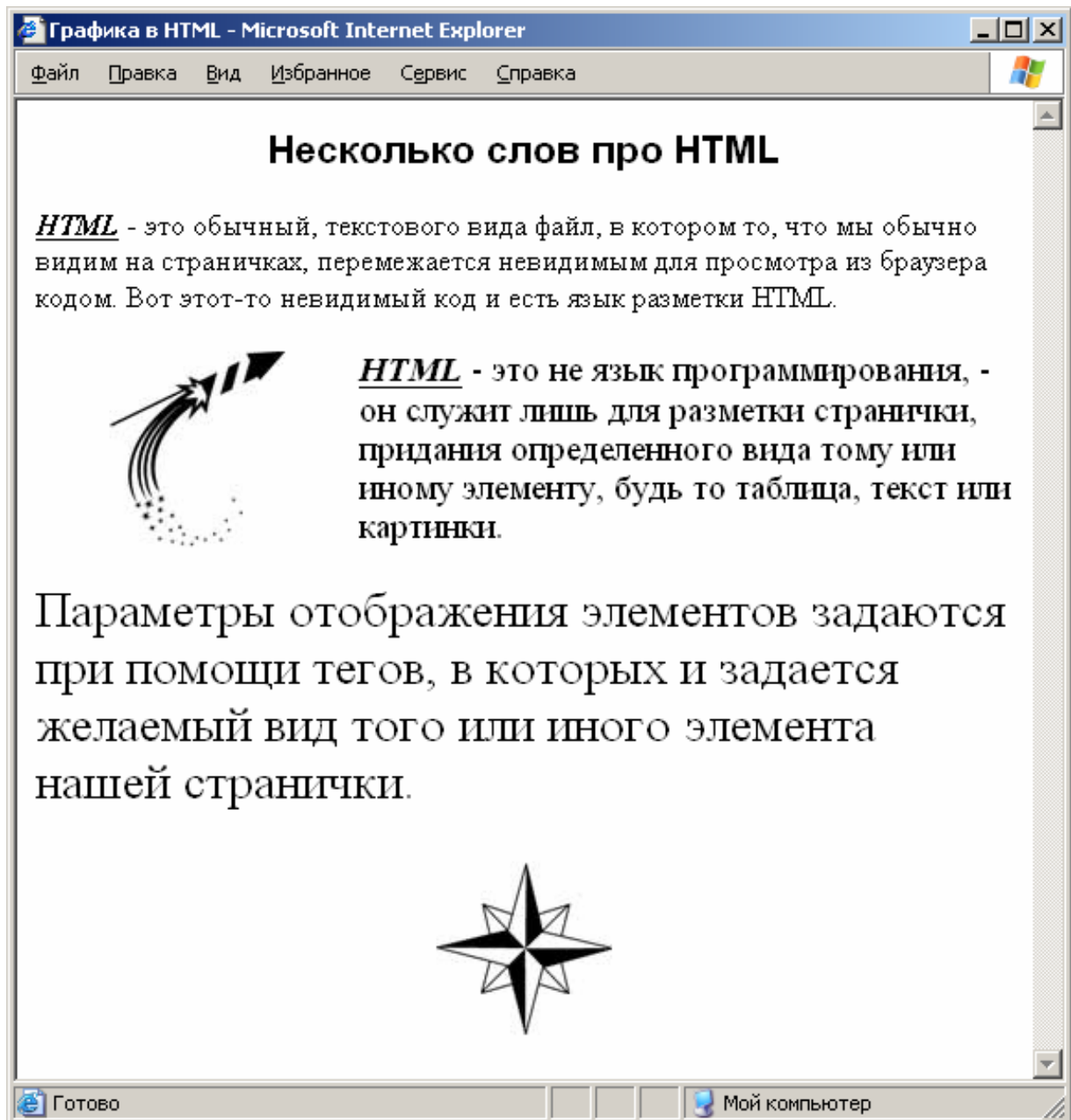
Контрольные задания

1. Создайте HTML-документ согласно образцу:



Примечания по выполнению. Для создания буквицы используйте рисунок с соответствующим выравниванием. Не забудьте про альтернативный текст к каждому рисунку.

2. Создайте HTML-документ по образцу (рисунки выберите на ваше усмотрение, измените их размеры, если это необходимо). У каждого рисунка задайте альтернативный текст.



Глава 4. Работа с таблицами

Существует два основных направления использования таблиц в HTML:

1. Отображение каких-либо данных в табличном виде.

В качестве данных таблицы может быть использовано все: параграфы, списки, заголовки, формы, рисунки и отформатированный текст.

2. Размещение всей информации на страничке внутри одной или нескольких таблиц.

Структуризация информации с помощью таблиц считается правилом хорошего тона в дизайне. Тексты, рисунки, ссылки размещаются в ячейках. Это значительно облегчает работу с позиционированием объектов относительно друг друга на странице.

Также таблицы помогают обойти ограничения HTML. Например, в HTML нельзя точно разместить элементы по горизонтали, но, разместив их в ячейках одной строки или столбца, можно выровнять положение элементов по горизонтали или вертикали соответственно.

Создание таблиц вручную - одна из наиболее сложных задач в HTML. Сначала вы формируете таблицу, затем первую строку, затем каждую ячейку с ее содержимым. Каждый элемент требует открывающего и закрывающего тэгов разметки документа. Если вы забыли один-единственный тэг, то вашу страницу никто не сможет прочитать.

Создание таблицы

Таблица описывается парными тэгами `<TABLE>` и `</TABLE>`:

```
<TABLE>  
тэги описания структуры таблицы  
</TABLE>
```

Тэг `<TABLE>` имеет параметры, с помощью которых можно изменять внешний вид таблицы:

- **border** - задает ширину границы таблицы (если `border=0`, то таблица отображается без границы);
- **bordercolor** – задает цвет границы таблицы;
- **width** - задает ширину всей таблицы (в пикселях или процентах от ширины окна просмотра);
- **height** - задает высоту всей таблицы (в пикселях или процентах от высоты окна просмотра);
- **bgcolor** – задает цвет фона таблицы;
- **background** – задает фоновый рисунок таблицы;
- **align** - расположение таблицы на странице (может принимать значения *left*, *center*, *right*);
- **cellpadding** - отступ текста от границ ячеек;

- **cellspacing** - отступ табличных объектов (ячейки, строки) друг от друга.

Структура таблицы

Кроме задания самой таблицы, нужно описать ее структуру.

При описании структуры таблицы используются 4 элемента, которые располагаются между тэгами `<TABLE>` и `</TABLE>`, - это **название**, **строка**, **ячейки-заголовки** и **ячейки с данными**.

Название является необязательным элементом и задается парными тэгами `<CAPTION></CAPTION>`

`<CAPTION>` Название таблицы `</CAPTION>`

У тэга `<CAPTION>` есть всего один параметр:

- **align** - задает положение названия таблицы относительно таблицы. Он может принимать значения:
 - **bottom** – название таблицы будет расположено в центре под таблицей;
 - **left** - название таблицы будет расположено над таблицей, слева;
 - **right** – название таблицы будет расположено над таблицей, справа;
 - **top** – название таблицы будет расположено над таблицей по центру (это значение задается по умолчанию).

Обязательный элемент **Строка таблицы** используется для формирования строк в таблице и задается парными тэгами `<TR></TR>`:

`<TR>`Строка таблицы`</TR>`

Тэг `<TR>` имеет 4 параметра:

- **align** – задает выравнивание внутри строки (возможные значения: **left**, **right**, **center**);
- **valign** – задает вертикальное выравнивание внутри строки (возможные значения: **top**, **bottom**, **middle**);
- **bgcolor** – задает цвет фона внутри строки;
- **background** – задает фоновый рисунок строки.

Внутри строки могут находиться ячейки с данными и ячейки-заголовки.

Ячейка-заголовок является необязательным элементом и служит для обозначения заголовка строки или столбца. По сути, это обычная ячейка таблицы, в которой используется более крупный и жирный шрифт, нежели в ячейках с данными.

Ячейка-заголовок задается парными тэгами `<TH></TH>`:

`<TH>`Заголовок столбца таблицы`</TH>`

Обязательный элемент **Ячейка таблицы** служит для хранения данных таблицы и задается парными тэгами `<TD></TD>`:

`<TD>`Ячейка с данными `</TD>`

Тэги **<TH>** и **<TD>** имеют несколько параметров:

- **align** – задает выравнивание внутри ячейки (*left, right, center*);
- **valign** – задает вертикальное выравнивание внутри ячейки (*top, bottom, middle*);
- **colspan** – растягивание ячейки на несколько столбцов;
- **rowspan** – растягивание ячейки-заголовка на несколько строк;
- **height** – задает высоту ячейки (в пикселях либо в процентах от высоты таблицы);
- **width** – ширина ячейки (в пикселях либо в процентах от ширины таблицы);
- **bgcolor** – задает цвет фона ячейки;
- **background** – задает фоновый рисунок ячейки;
- **nowrap** – наличие или отсутствие этого параметра обозначает запрет или разрешение на перенос строк внутри ячейки.

Упражнение 4.1

Создайте HTML-документ *tables.html* с названием странички «Работа с таблицами в HTML». Поместите в этот документ простейшую таблицу, имеющую название «Пример простой таблицы» и состоящую из трех столбцов с заголовками и трех строк (для наглядности можно использовать в коде отступы):

```
<TABLE border=1>
<CAPTION>Пример простой таблицы</CAPTION>
<TR>
  <TH>Заголовок1</TH>
  <TH>Заголовок 2</TH>
  <TH>Заголовок 3</TH>
</TR>
<TR>
  <TD>ячейка A</TD>
  <TD>ячейка B</TD>
  <TD>ячейка C</TD>
</TR>
<TR>
  <TD>ячейка D</TD>
  <TD>ячейка E</TD>
  <TD>ячейка F</TD>
</TR>
<TR>
  <TD>ячейка G</TD>
  <TD>ячейка H</TD>
  <TD>ячейка J</TD>
</TR>
</TABLE>
```

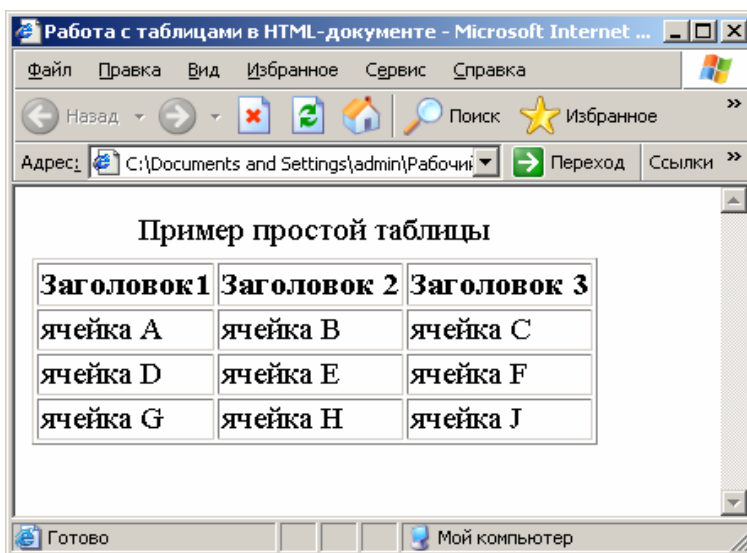


Рис. 4.1. Пример простейшей таблицы в HTML

и посмотрите ее в браузере (рис. 4.1)

Задание 4.1

Измените внешний вид таблицы, созданной в упр. 4.1:

- используя тэг , отформатируйте текст названия таблицы: *курсивное* начертание, **красный** цвет текста;
- измените положение названия таблицы - над таблицей справа;
- используя параметр **bgcolor**, задайте цвета ячеек, пусть все ячейки будут различного цвета;
- используя тэг , задайте различные цвета шрифта в ячейках таблицы.

Параметры rowspan, colspan

Для объединения группы ячеек в таблице используются специальные параметры: **rowspan** («растягивает» ячейку таблицы на указанное количество строк) и **colspan** (растягивает ячейку таблицы на указанное количество столбцов)

Параметр **rowspan** строк. Параметр **colspan** растягивает ячейку

Упражнение 4.2

Приведите таблицу, созданную в упр. 4.1:

Заголовок 1	Заголовок 2	Заголовок 3
Ячейка А	Ячейка В	Ячейка С
Ячейка D	Ячейка Е	Ячейка F
Ячейка G	Ячейка H	Ячейка J

К виду:

Заголовок 1	Заголовок 2	Заголовок 3
Объединенные ячейки А, D, G	Ячейка В	Ячейка С
	Ячейка Е	Ячейка F
	Ячейка H	Ячейка J

Для этого:

- Нам нужно «растянуть» ячейку А на три строки.
- Откройте код таблицы.
- При описании таблицы у тэга, описывающего ячейку А, укажите параметр **rowspan=3** (то есть «растянуть» на 3 строки):

```
<TD rowspan=3>Объединенные ячейки А, D, G</TD>.
```

- Растянув таким образом ячейку А, мы уже не нуждаемся в ячейках D и G – убираем описание этих ячеек.
- HTML-код будет выглядеть так (изменения выделены жирным шрифтом):

```
<TABLE border=1>
<CAPTION>Пример простой таблицы</CAPTION>
<TR>
  <TH>Заголовок1</TH>
  <TH>Заголовок 2</TH>
```

```

    <TH>Заголовок 3</TH>
  </TR>
  <TR>
    <TD rowspan=3>Объединенные ячейки A, D, G</TD>
    <TD>ячейка B</TD>
    <TD>ячейка C</TD>
  </TR>
  <TR>
    <TD>ячейка E</TD>
    <TD>ячейка F</TD>
  </TR>
  <TR>
    <TD>ячейка H</TD>
    <TD>ячейка J</TD>
  </TR>
</TABLE>

```

- Посмотрите страничку в окне браузера.

Задание 4.2

Используя параметр **colspan**, приведите таблицу в HTML-документе *tables.html*:

Заголовок 1	Заголовок 2	Заголовок 3
Объединенные ячейки A, D, G	Ячейка B	Ячейка C
	Ячейка E	Ячейка F
	Ячейка H	Ячейка J

К виду:

Объединенные ячейки A, D, G	Ячейка B	Ячейка C
	Объединенные ячейки E и F	
	Ячейка H	Ячейка J

Границы (бордюры) таблицы

Мы уже упомянули параметры **border** и **bordercolor** тэга <TABLE>.

Параметр **border** устанавливает ширину бордюра (рамки) таблицы и может принимать любое значение от 0 до 10. Ширина бордюра=0 означает его отсутствие. Если значение параметра border опускается, то border=0.

Параметр **bordercolor** задает цвет границы таблицы.

Упражнение 4.3

Создайте HTML-документ *pic.html* и поместите в него рисунок, окруженный красной рамкой.

Для этого:

- Так как у тэга нет параметра, задающего цвет границы, то поместим рисунок в таблицу с одной-единственной ячейкой и зададим красный цвет границы таблицы.

Задание 4.3

У таблицы в HTML-документе *tables.html* задайте малиновый цвет границы толщиной 3 пикселя.

Параметр width, height

Когда вы создаете таблицу, то ширина столбца оказывается равной самому широкому элементу этого столбца. При желании можно установить для столбца точную ширину. Вот здесь и пригодится параметр **width**. При работе с высотой используется оператор **height**.

Упражнение 4.4

Создайте HTML-документ *table_sizes.html* со следующей таблицей:

Таблица в 50% ширины от экрана. Первый столбец в 80% ширины таблицы	Ячейка 2
Ячейка 3	Ячейка 4

Для этого:

- У тэга `<TABLE>` укажите параметр `width=50%`:

```
<TABLE border=1 width=50%>
```

- У тэга `<TD>` первой ячейки нужного столбца укажите параметр `width=80%`:

```
<TD width=80%>
```

- Посмотрите страничку в окне браузера.

Задание 4.4

Измените таблицу, созданную в упр. 4.4: задайте ширину первого столбца 30%, высоту таблицы 50% от экрана. Внесите в ячейки соответствующий поясняющий текст.

Вертикальное и горизонтальное выравнивание внутри ячейки

Для задания выравнивания элементов внутри ячейки у тэга `<TH>` или `<TD>` используются параметры **align** - для задания горизонтального выравнивания внутри ячейки (возможные значения *left, right, center*) и **valign** – для задания вертикального выравнивания внутри ячейки (возможные значения *top, bottom, middle*).

Упражнение 4.5

В HTML-документе *tables_sizes.html*, созданном в упр. 4.4, в ячейках таблицы установите вертикальное выравнивание по середине и горизонтальное – по центру.

Для этого:

- У каждого тэга ячейки `<TD>` укажите параметры `valign=middle` и `align=center`:

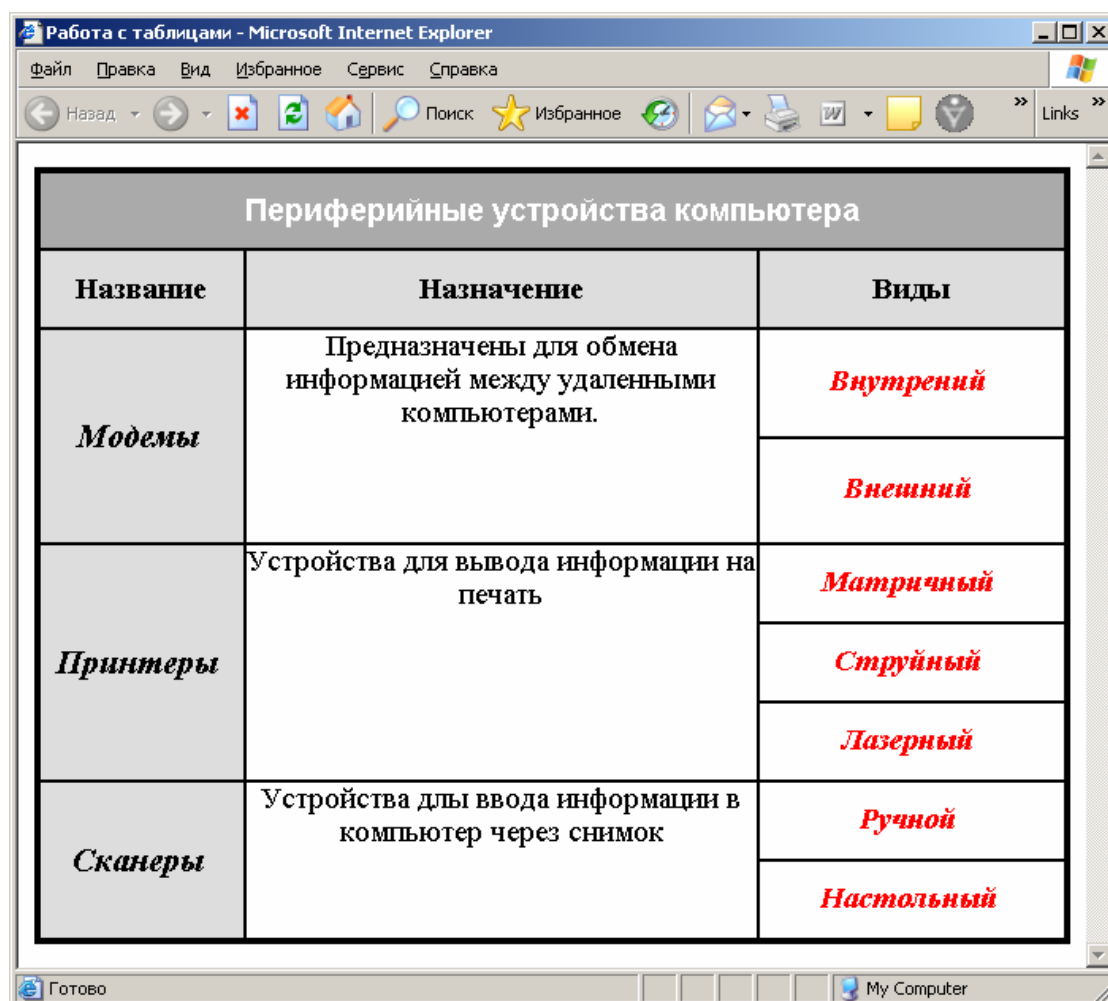
`<TD valign=middle align=center>`

Вопросы для повторения

1. Как можно использовать таблицы в HTML (два направления)?
2. Назовите основные элементы в структуре таблицы. Какие элементы являются обязательными и необязательными?
3. Перечислите параметры, которые применяются при объединении нескольких ячеек. Приведите примеры.

Контрольные задания

1. Создайте HTML-документ с таблицей по образцу:

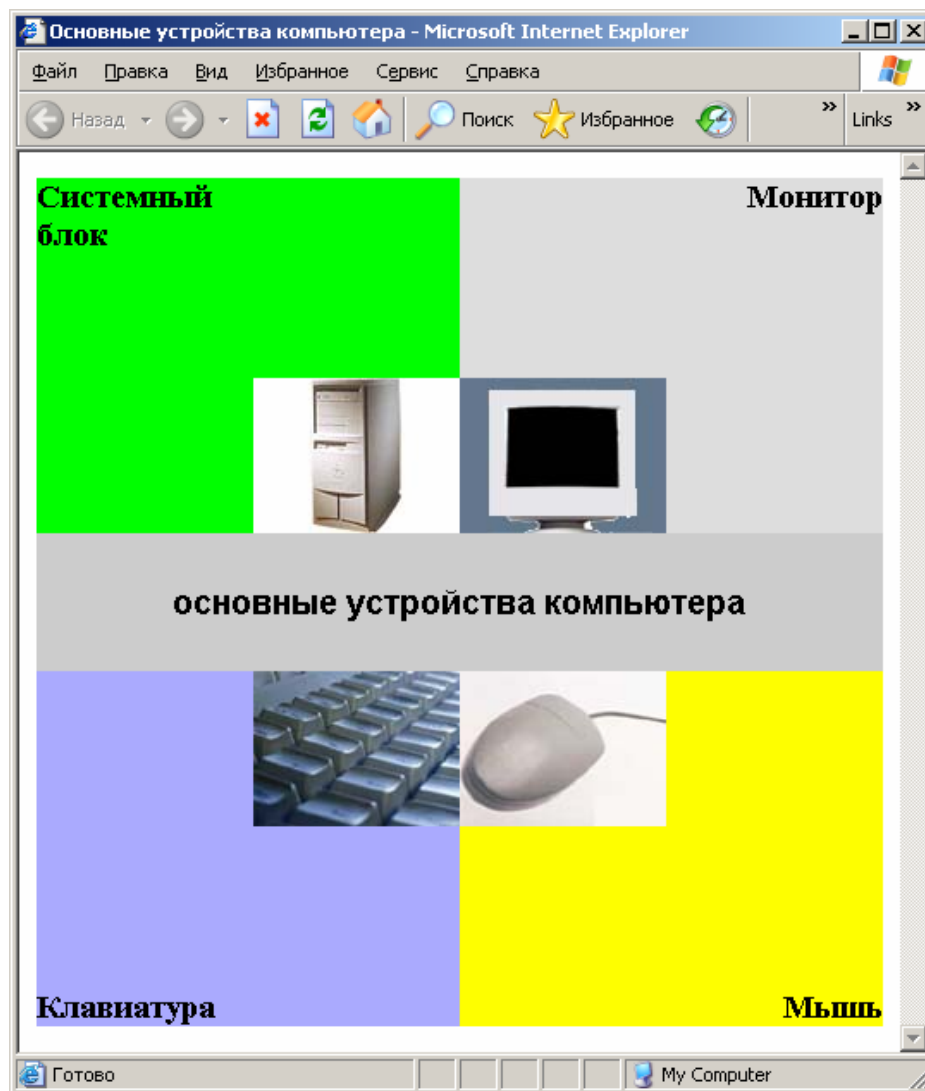


The screenshot shows a web browser window titled "Работа с таблицами - Microsoft Internet Explorer". The address bar shows a local file path. The main content area displays a table with the following structure:

Периферийные устройства компьютера		
Название	Назначение	Виды
<i>Модемы</i>	Предназначены для обмена информацией между удаленными компьютерами.	<i>Внутренний</i>
		<i>Внешний</i>
<i>Принтеры</i>	Устройства для вывода информации на печать	<i>Матричный</i>
		<i>Струйный</i>
		<i>Лазерный</i>
<i>Сканеры</i>	Устройства для ввода информации в компьютер через снимок	<i>Ручной</i>
		<i>Настольный</i>

Используйте различные цвета фона и текста.

2. Создайте HTML-документ по образцу:



Примечание по выполнению. *Картинки и подписи к картинкам размещены в разных ячейках с различным выравниванием.*

Глава 5. Гиперссылки

Часть текста, название, участок изображения и пр. могут иметь ссылки на другой текст внутри этого же документа или на другой документ в сети Интернет. Такие ссылки называются гипертекстовыми, или гиперсвязью. Существуют ссылки текстовые и графические (когда ссылкой является рисунок). По умолчанию текстовая ссылка выделяется синим цветом и подчеркиванием, а графическая – синей рамкой.

Но цвет ссылок на страничке можно устанавливать и самим через специальные параметры тэга `<BODY>`:

- **link** - цвет ссылки, еще не посещенной;
- **alink** - цвет активной ссылки;
- **vlink** - цвет ссылки, уже посещенной;

Например,

```
<BODY link="blue" alink="green" vlink="red">
```

задает синий цвет еще не посещенным ссылкам, зеленый – активным, красный – уже посещенным.

Создание гиперссылки

Для добавления гиперссылки в HTML-документ используют парные тэги `<A>` и ``:

```
<A href="путь к документу">текст ссылки (или рисунок)</A>
```

Обязательный параметр **href** тэга `<A>` определяет адрес документа, на который будет совершен переход, если щелкнуть мышью по объекту, находящемуся между тэгами. Адресом может быть либо локальный документ, либо страничка в сети Интернет.

Помимо **href** тэг `<A>` имеет ряд других, необязательных, параметров:

- **title** – текст подсказки, которая всплывает при наведении на ссылку;
- **target** – наличие или отсутствие этого параметра определяет, как открыть документ, на который ведет ссылка;
- **name** – используется при задании ссылки внутри текущего документа (метки¹).

Упражнение 5.1

Создание текстовой ссылки.

Создайте HTML-документ *links.html* с заголовком странички «Ссылки в HTML-документе» с текстом «Я учусь в ВятГГУ». Название университета оформите ссылкой, ведущей на сайт www.vspu.kirov.ru.

Для этого:

- Название университета заключите между тэгами `<A>` и ``. В параметре **href** тэга `<A>` укажите полный адрес сайта:

¹ Более подробно о метках речь пойдет далее.

`ВятГГУ`



Рис. 5.1. Изменение внешнего вида курсора при наведении на ссылку

- Проверьте работу ссылки: ссылка окрашена в синий цвет и подчеркнута. При наведении курсора мыши на ссылку он меняется, а в строке состояния отображается адрес ссылки (рис. 5.1).

Упражнение 5.2

У HTML-документа *links.html*, созданного в упр. 5.1, задайте следующие настройки ссылок: текст непосещенных ссылок окрашивается синим цветом, текст активных ссылок – зеленым, посещенных – красным.

Для этого:

- Откройте код странички *links.html*.
- У тэга `<BODY>` задайте нужные значения соответствующих параметров:

`<BODY link="blue" alink="green" vlink="red">`

- Проверьте настройки ссылок:
 - кликнув по ссылке, вы автоматически перейдете на документ, на который вела эта ссылка. Заметьте, что в момент активизации ссылки (то есть в момент клика) ссылка меняла цвет на зеленый;
 - вернитесь на исходный документ, нажав кнопку «Назад» на панели инструментов окна браузера (или просто нажмите клавишу «BackSpace»). Ссылка все еще остается активной – следовательно зеленой;
 - теперь щелкните по любому месту в вашем HTML-документе: ссылке присваивается статус посещенной – ссылка станет красной.

Задание 5.1

Добавьте у ссылки, созданной в упр. 5.1, подсказку «Перейти на сайт ВятГГУ». Проверьте результат в окне браузера: при наведении на ссылку должна появляться подсказка.

Упражнение 5.3

Создание графической ссылки.

Добавьте в HTML-документ `links.html` графическую ссылку, ведущую на сайт www.afisha.kirov.ru (рис. 5.2).

Для этого:

- В любом графическом редакторе (например, «Paint») создайте рисунок



- Создайте ссылку на сайт www.afisha.kirov.ru, но вместо текста ссылки поместите созданный рисунок¹:

```
<A href="http://www.afisha.kirov.ru/">  
<IMG src="afisha.jpg" alt="перейти на  
сайт Афиша.Киров.RU">  
</A>
```

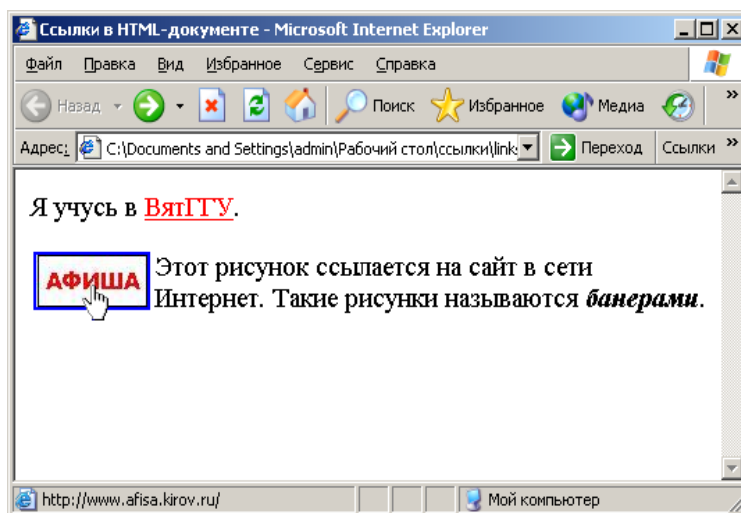


Рис. 5.2. Графическая ссылка в HTML-документе

- Рядом со ссылкой разместите комментирующий текст.
- Загрузите страничку в окне браузера.

Задание 5.2

Уберите ссылочную рамку вокруг рисунка.

Примечание. Используйте параметр `border` у тэга ``.

Упражнение 5.4

Создание гиперсвязи между локальными страничками.

В HTML-документе `links.html` создайте ссылку, ведущую на страничку с картинкой.

Для этого:

- Создайте HTML-документ `picture.html` с картинкой с текстом «Вы попали на страничку с картинкой!».
- В HTML-документ `links.html` добавьте ссылку на эту страничку:

```
Щелкнув <A href="picture.html">здесь</a> мышью, вы можете попасть  
на файл с картинкой!
```

- Откройте этот документ в браузере и проверьте работу ссылки.

¹ Подробнее о вставке рисунка читайте в гл. 3 «Графика в HTML».

Задание 5.3

Усовершенствуйте переход между страничками, созданный в упр. 5.4, – на страничке с картинкой добавьте ссылку «Назад», по которой можно вернуться на исходный HTML-документ.

Параметр target

Когда пользователь щелкает по ссылке, документ открывается в текущем окне. Иногда удобнее открывать ссылочный документ в отдельном окне. Это можно осуществить, щелкнув по ссылке правой кнопкой мыши и в контекстном меню выбрав команду **Открыть в новом окне** (рис. 5.3).

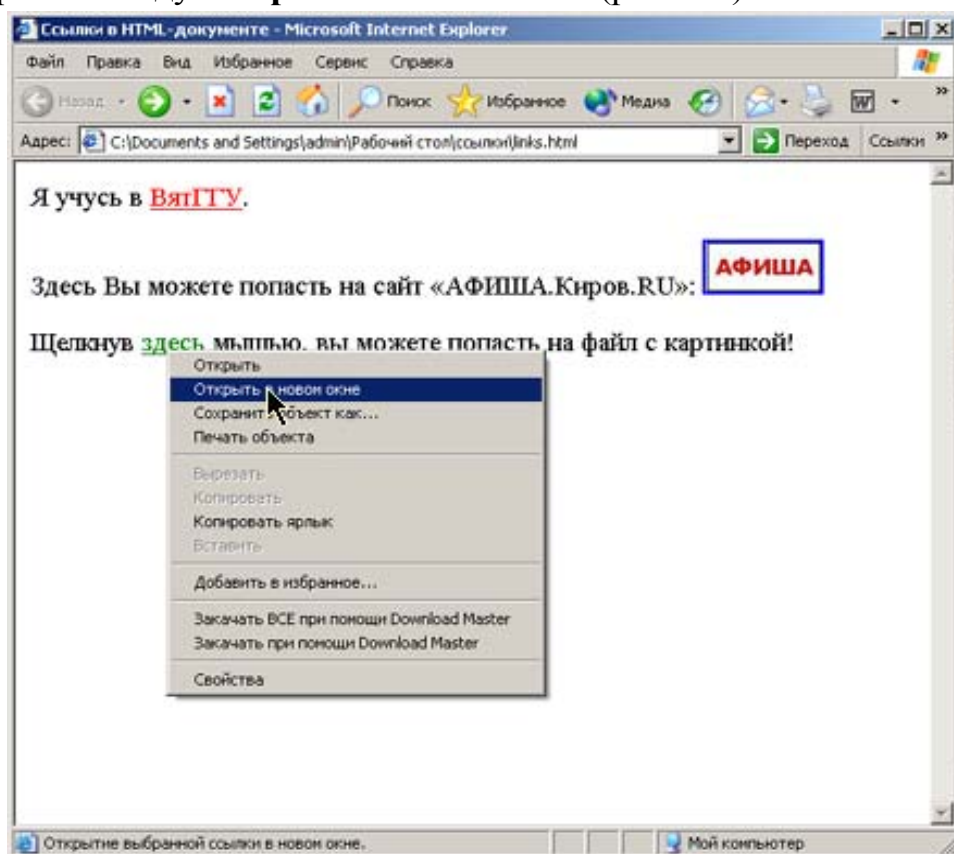


Рис. 5.3. Открытие документа по ссылке в новом окне
через контекстное меню

Можно настроить открытие документа в новом окне средствами HTML. Для этого нужно параметру **target** присвоить значение «**_blank**». Например,

```
<A href="http://www.vspu.kirov.ru" target="_blank">ВятГУ</A>
```

Параметр target может иметь еще следующие значения:

- **_self** – документ будет загружаться в текущем окне, то есть в том же, что и ссылка на этот документ. По умолчанию присваивается это значение;

- ***_parent*** – используется при работе с фреймами¹: документ будет загружаться во фрейм-родитель, если фреймов нет, то этот параметр работает как ***_self***;
- ***_top*** – используется при работе с фреймами: документ загружается в полном окне без фреймов, если фреймов нет, то этот параметр работает как ***_self***.

Метки

Обычно, когда загружается страничка, пользователь автоматически перемещается в начало странички. Для перемещения по ссылкам в конкретную часть текущей или другой странички используются метки.

Место, в которое пользователь будет перемещаться, щелкнув по ссылке, метится тэгом `<A>` с параметром `name`:

```
<A name="имя метки">текст метки</A>
```

Имя метки может быть любое: числа и буквы.

Ссылка на метку оформляется следующим образом:

```
<A href="имя документа#имя метки">текст ссылки</A>
```

Примечание. Если ссылка ведет на место в этом же HTML-документе, то имя документа в параметре `href` можно не указывать.

Пользователь, кликнув на текст ссылки, попадает в то место странички, где расположен текст метки.

Упражнение 5.5

Создайте ссылку внутри HTML-документа `links.html`, щелкнув по которой, пользователь перемещается в конец документа.

Для этого:

- Откройте код странички `links.html`.
- В конце тела странички создайте метку с именем `finish`:

```
<A name="finish">конец документа</A>
```

• В начале документа поместите ссылку с текстом «в конец документа», ведущей на метку с именем `finish`:

```
<A href="links.html#finish">в конец документа</A>
```

- Заметим, что имя документа `links.html` можно опустить, потому что метка находится в этом же документе.
- Проверьте работу ссылки, загрузив страничку в окне браузера.

¹О фреймах будет рассказано в следующей главе.

Размещение адреса электронной почты

Также с web-странички можно отправлять письмо по электронной почте. Ссылка на электронный адрес оформляется с помощью тэга <A> с указанием в параметре href специального слова **mailto:**

Например:

Пишите письма на:vasenina@mail.ru

Когда пользователь щелкнет по этой ссылке, на компьютере запускается почтовая программа и создается письмо с уже заполненным полем «Кому» (рис. 5.4).

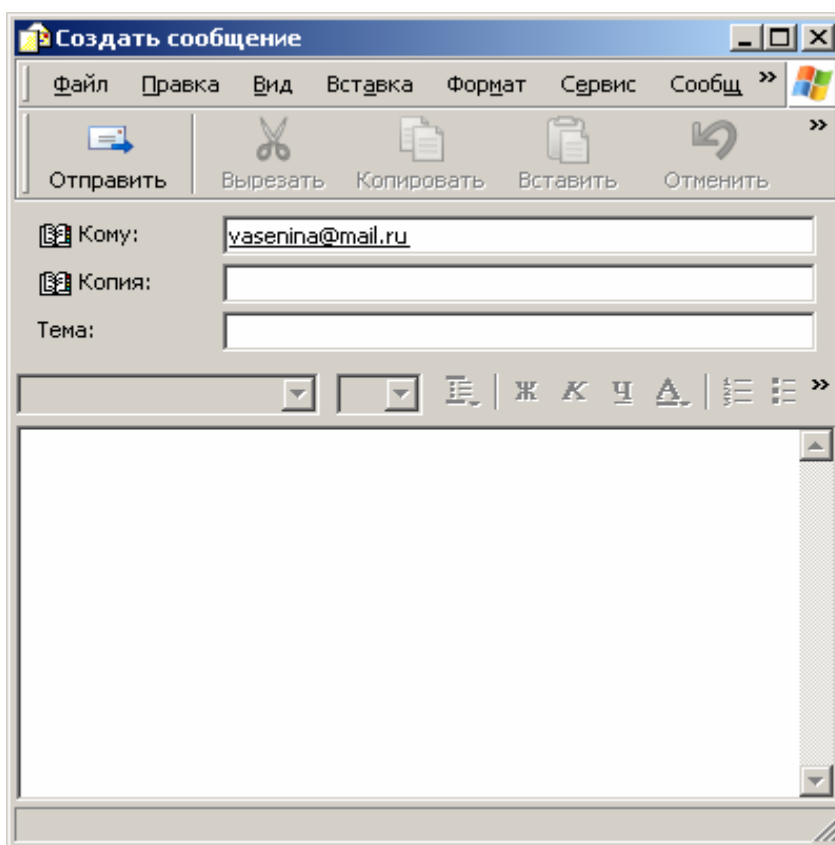


Рис. 5.4. Окно создания письма, появившееся после активизации ссылки с адресом электронной почты

Совет будущему HTML-редактору. Чтобы при создании письма автоматически заполнялась и тема письма, нужно в конец адреса ссылки добавить «?subject=» и указать тему письма. Например,

Все вопросы пишите на:

inform@mail.ru.

Упражнение 5.6

Поместите на свою страничку ссылку на свой адрес электронной почты. Если у вас нет зарегистрированного адреса, то для выполнения упражнения придумайте его сами. Формат адреса электронной почты:

имя_пользователя@адрес_почтового_сервера

Карта ссылок

В сети Интернет можно встретить рисунок, на котором в разных его частях размещены сразу несколько ссылок. Это осуществляется с помощью специального механизма **Карта ссылок** (MAP).

Этот механизм включает в себя сразу несколько тэгов: **** с атрибутом **usemap**, **<MAP>** с атрибутом **name**, **<AREA>** с атрибутами **shape**, **coords**, **href** и закрывающий тэг **</MAP>**.

Перед вами пример меню на сайте www.shakers.kirov.ru, которое оформлено в виде карты:

```
<IMG border="0" src="menu.jpg" usemap="#menu" width="113" height="268">
<MAP name="menu">
<AREA shape="rect" alt="История" coords="15,18,98,48" href="history.html">
<AREA shape="rect" alt="Афиша" coords="6,44,90,74" href="afisha.html">
<AREA shape="rect" alt="Новости" coords="13,76,108,105" href="news.html">
<AREA shape="rect" alt="Фото" coords="10,105,94,135" href="photo.html">
<AREA shape="rect" alt="Гостевая" coords="12,173,102,200" href="guest.html">
<AREA shape="rect" alt="Контакты" coords="10,200,90,232" href="contacts.html">
</MAP>
```

Остановимся поподробнее на синтаксисе Карты.

Вначале нам нужно задать рисунок, который будет использоваться в качестве карты (*menu.jpg* в нашем примере). Кроме того, рисунку нужно сообщить, карту с каким именем будут на него накладывать (**usemap="#menu"**), имя указывается обязательно с символом #:

```
<IMG border="0" src="menu.jpg" usemap="#menu">
```

Затем начинаем описание самой карты ссылок:

- 1) задаем имя карте через атрибут **name** тэга **<MAP>** (имя может быть любое):

```
<MAP name="menu">
```

- 2) используя тэг **<AREA>** (область), начинаем распределять места на рисунке под ссылки:

```
<AREA shape="форма области" coords="координаты области" href="адрес ссылки"
title="подсказка">
```

Тэг **<AREA>** имеет ряд параметров:

- **shape** - этот параметр задает форму активной области на карте, он может принимать значения:
 - “**rect**” - прямоугольник,
 - “**circle**” - круг,
 - “**poly**” - многоугольник,
 - “**default**” - задается область, покрывающая незаданную часть рисунка (должна задаваться самой последней);
- **coords** - этот параметр задает координаты активной области на карте. Количество координат обусловлено параметром **shape**:

- `shape="rect" coords="X1,Y1,X3,Y3"` - задаются координаты верхнего левого и правого нижнего угла прямоугольника,
- `shape="circle" coords="X,Y,R"` - где X,Y,R - координаты центра круга и его радиус,
- `shape="poly" coords="X1,Y1,X2,Y2,X3,Y3"...` – задаются координаты вершин многоугольника,
- `shape="default"` – не требуется задания координат, определяет вообще все точки и должно располагаться последним в списке областей.
- **href** - этот параметр определяет адрес ссылки;
- **alt** - этот параметр задает альтернативный текст для каждой из активных областей изображения;
- **title** - этот параметр задает текст подсказки, которая всплывает при наведении курсора на данный элемент

Упражнение 5.7

Создайте HTML-документ *map.html*. Создайте в этом документе меню из двух ссылок, ведущих на странички *picture.html* (страничка с картинкой) и *table.html* (страничка с таблицей). Если страничек нет, создайте их (не забудьте про названия и заголовки). Меню реализуйте с помощью карты.

Для создания карты:

- В любом графическом редакторе (например, «Paint») создайте рисунок, который будет служить меню на страничке. Например,



- Определите размер вашего рисунка. Например, высота=75 пикселей, ширина=80 пикселей.
- Определите области ссылок. Для нашего примера форма области – прямоугольник, для первой ссылки определяется верхняя половина рисунка, для второй – нижняя половина.
- Опишите карту в коде странички:

```
<IMG border="0" src="menu.jpg" usemap="#menu" width="80" height="75">
<MAP name="menu">
<AREA shape="rect" alt="Картинка" coords="0,0,80,37" href="picture.html">
<AREA shape="rect" alt="Таблица" coords="0,38,80,75" href="tables.html">
</MAP>
```

- Проверьте работу карты: перейдите по ссылкам на странички.

Вопросы для повторения

1. Что такое гиперссылка? Назовите два вида гиперссылок.
2. Какой тэг задает ссылку? Перечислите основные и вспомогательные параметры этого тэга.
3. Что такое метка? Как ее задать?
4. Как задать ссылку на адрес электронной почты?
5. Какими способами можно открыть ссылку в новом окне?
6. Что такое карта ссылок?

Контрольные задания

1. С помощью меток реализуйте сноски в HTML-документе: если пользователь находит в тексте сноску, то, щелкая по ней, перемещается в объяснение этой сноски, которое находится в конце документа. Кроме того, если навести на номер сноски курсор мыши, то текст сноски появляется в подсказке.

Примечание. Яркий пример можно найти на личном сайте М. И. Ненашева, профессора кафедры философии ВятГГУ (www.nenashev.kirov.ru), например, на странице www.nenashev.kirov.ru/monografies/02/index.shtml?pred

2. Создайте HTML-документ, содержащий ссылки на странички с картинкой, с таблицей, со списком. При наведении курсора мыши на ссылку должна всплывать подсказка. Отформатируйте странички: добавьте цвет фона, цвет текста, все странички оформите в едином стиле - одинаковый шрифт, одинаковый фон. Каждая страничка имеет свое название, заглавие и ссылку «назад», при нажатии на которую пользователь возвращается на главную страничку.

Глава 6. Фреймы

Фреймы (*frame* – структура, рамка) позволяют разбить окно браузера на несколько независимых окон, в каждом из которых будет отображаться один HTML-документ.

Использование фреймов позволяет размещать статическую информацию (например, главное меню сайта, графический логотип фирмы, copyright, набор управляющих кнопок) в одном фрейме, а изменяющуюся - в другом.

Используя фреймы, позволяющие разбивать web-страницы на множественные подокна, в некоторых случаях вы можете значительно улучшить внешний вид и функциональность ваших страничек.

Фреймы имеют следующие возможности:

- Каждый фрейм имеет свой URL, что позволяет загружать его независимо от других фреймов.
- Каждый фрейм может иметь собственное имя (параметр *name*), позволяющее обращаться к нему из другого фрейма.
- Размер фрейма может быть изменен пользователем прямо на экране при помощи мыши (если это не запрещено указанием специального параметра).

Но тут же следует заметить, что система фреймирования не всегда правильно воспринимается большинством поисковых систем. Практика показывает, что количество посещений страничек с фреймами значительно меньше, чем тех же страничек, с совершенно тем же содержимым, но без использования фреймов.

Создание web-странички на основе фреймов

Создание web-странички на основе фреймов осуществляется в два этапа:

1. Создается HTML-файл (главный), в котором задаются размеры и количество фреймов, а также имена файлов, соответствующих фреймам, и некоторые атрибуты для каждого фрейма.
2. Создаются отдельные HTML-документы, которые будут отображаться во фреймах.

Главный документ – специфичный вид HTML-документа. Он не содержит тэга `<BODY>`, он только описывает фреймы, которые будут содержать информацию. В нем осуществляется разбиение на фреймы.

Структура главного документа:

```
<HTML>
<HEAD>...</HEAD>
  <FRAMESET>
    <FRAME...>
    <FRAME...>
    ...
  </FRAMESET>
</HTML>
```

Между парными тэгами **<FRAMESET>** и **</FRAMESET>** создаётся фреймовая структура. Эти тэги используются вместо тэгов **<BODY>** и **</BODY>**.

Тэг **<FRAMESET>** имеет несколько параметров:

- **rows** – определяет количество фреймов в окне браузера по горизонтали и соотношение размеров этих фреймов;
- **cols** – определяет количество фреймов в окне браузера по вертикали и соотношение размеров этих фреймов.

Например,

```
<FRAMESET cols="100,*">
```

разбивает окно браузера на два вертикальных фрейма, ширина первого 100 пикселей, второго – вся оставшаяся ширина окна. Размер можно задавать в пикселях или процентах.

Предупреждение. Совместное использование параметров **rows** и **cols** может привести к непредсказуемым результатам. Например, код

```
<FRAMESET rows="50%,50%" cols="50%,50%">
```

может привести к ошибочной ситуации.

- **frameborder** – определяет наличие рамки, обрамляющей фреймы (принимает значения “yes” или “no”, по умолчанию **frameborder=“yes”**);
- **border** – задает толщину рамки, обрамляющей фреймы (этот параметр нужен в том случае, если параметр **frameborder=“yes”**, измеряется в пикселях);
- **bordercolor** – задает цвет рамки, обрамляющей фрейм.

С помощью тэга **<FRAME>** задаются атрибуты для каждого фрейма персонально.

Тэг **<FRAME>** имеет следующие параметры:

- **src** – обязательный параметр, задает путь к файлу, который отображается в данном фрейме;
- **name** – задает имя фрейма. Имя необходимо для того, чтобы впоследствии можно было в этом же фрейме отображать другие файлы (подробнее о нем написано ниже);
- **scrolling** – определяет наличие полосы прокрутки во фрейме (принимает значения «yes» или “no”, по умолчанию **scrolling=“yes”**);
- **noresize** – отсутствие или наличие этого параметра говорит о том, можно или нельзя изменять размеры фрейма;
- **marginwidth**, **marginheight** – задают отступ содержимого внутри фрейма по горизонтали и вертикали соответственно (измеряется в пикселях).

Файлы, которые отображаются во фреймах, – это обыкновенные HTML-документы со стандартной структурой. Они могут содержать рисунки, таблицы, текст.

Упражнение 6.1

Создайте web-страничку, содержащую три горизонтальных фрейма: первый фрейм содержит таблицу, второй – рисунок, третий – отформатированный текст.

Для этого:

- Создайте файл главного документа – *frames.html*:

```
<HTML>
<HEAD><TITLE> Пример окна с фреймами </TITLE></HEAD>
<FRAMESET rows="30%,40%,*">
<FRAME name="table" src="table.html">
<FRAME name="picture" src="picture.html">
<FRAME name="text" src="text.html">
</FRAMESET>
</HTML>
```

- Создайте файлы-фреймы *table.html*, *picture.html*, *text.html*. Занесите в них соответствующую информацию.
- Просмотрите файл главного документа в окне браузера (рис. 6.1).

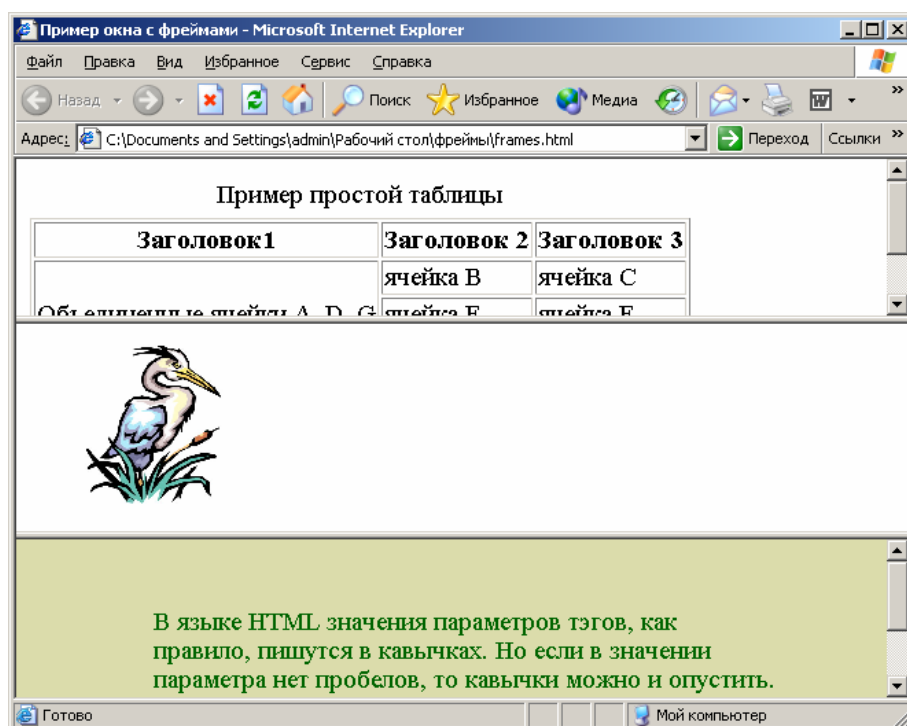


Рис. 6.1. Страничка с тремя горизонтальными фреймами

Задание 6.1

Измените расположение фреймов с горизонтального на вертикальный.

Упражнение 6.2

Удалите полосу прокрутки у первого фрейма (рис. 6.2). Для этого у тэга `<FRAME>`, описывающего первый фрейм, добавьте параметр `scrolling="no"`.

Передвигая границы фреймов, измените их размеры так, чтобы стала видна таблица полностью.

Задание 6.2

С помощью параметра `noresize` тэга `<FRAME>` запретите возможность изменения размеров первого фрейма. Проверьте выполнение этого запрета – попробуйте передвинуть границу фрейма.

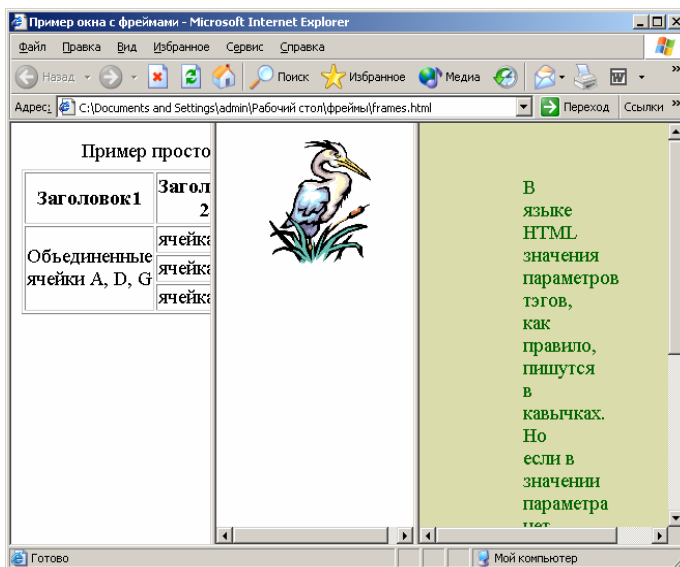


Рис. 6.2. Отсутствие полосы прокрутки у первого фрейма

Задание 6.3

С помощью параметров `bordercolor` и `border` тэга `<FRAMESET>` у границы между фреймами задайте синий цвет и толщину в 5 пикселей.

Задание 6.4

С помощью параметра `frameborder` тэга `<FRAMESET>` удалите границы между фреймами (рис. 6.3).

Задание 6.5

С помощью параметров `marginwidth`, `marginheight` тэга `<FRAME>` в первом фрейме задайте горизонтальный отступ 30 пикселей, вертикальный – 50 пикселей.

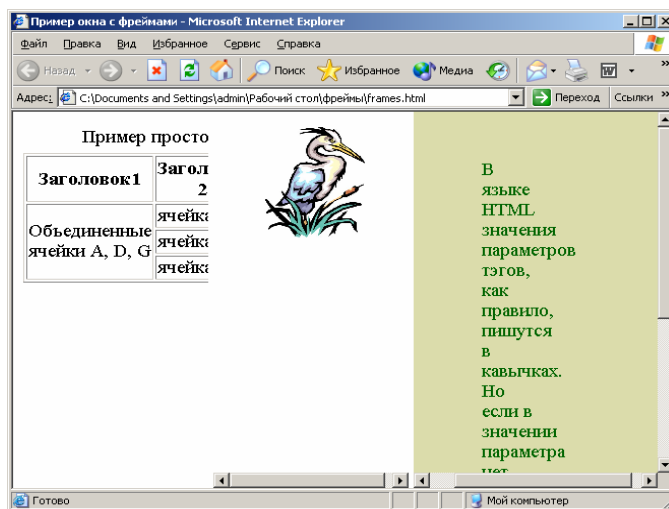


Рис. 6.3. Страницка с фреймами без границ

Упражнение 6.3

Создайте страничку *index.html*, содержащую два фрейма, первый представляет собой меню и отображает содержимое файла *menu.html*, второй – отображает содержимое в зависимости от выбранной ссылки в меню (рис. 6.4).

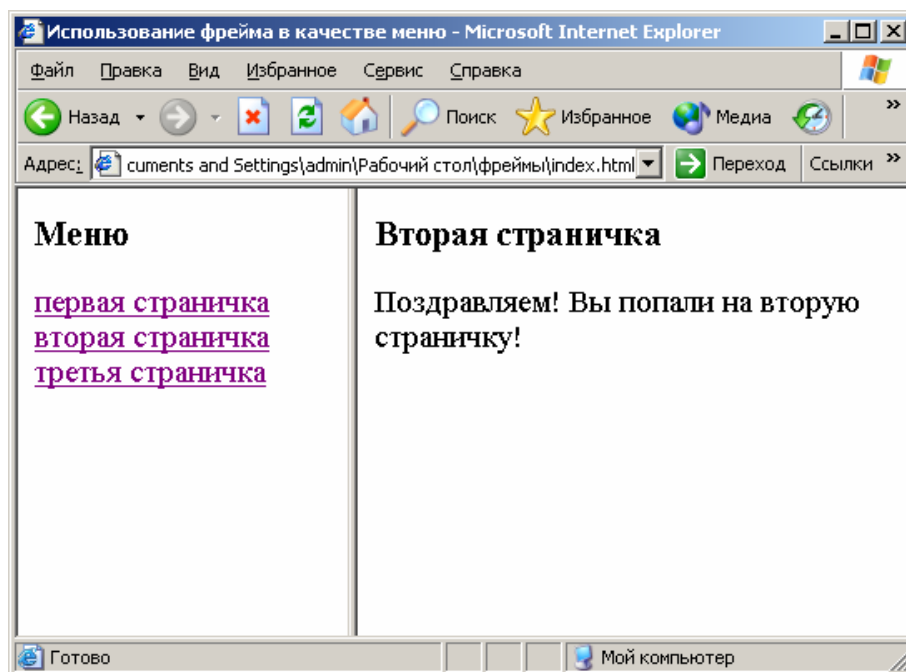


Рис. 6.4. Использование первого фрейма в качестве меню

Для этого:

- Создайте главный документ *index.html*.
- Проанализируем его структуру. Главный документ представляет собой два вертикальных фрейма: левый - для меню (пусть его ширина - 200 пикселей) и правый - для отображения выбранных в меню файлов. Следовательно, при описании тэга `<FRAMESET>` будем использовать параметр `cols`:

```
<FRAMESET cols="200,*">
```

- В левом фрейме будет отображаться содержимое файла *menu.html*:

```
<FRAME src="menu.html">
```

- Изначально в правом фрейме будет отображаться страничка с приветствием:

```
<FRAME src="greet.html">
```

- Так как в правом фрейме содержимое будет изменяться в зависимости от выбранного пункта меню, присвоим ему имя (например, *content*) для дальнейшего обращения из фрейма-меню:

```
<FRAME name="content" src="greet.html">
```

- В результате выполнения всех действий структура главного HTML-документа будет выглядеть так:


```
<HTML>
<HEAD>
<TITLE> Использование фрейма в качестве меню</TITLE>
</HEAD>
<FRAMESET cols="200,*">
<FRAME src= "menu.html">
<FRAME name="content" src= "greet.html">
</FRAMESET>
</HTML>
```

- Начинаем создавать файлы-фреймы.
- Пусть файл *menu.html* содержит три ссылки – «первый фрейм», «второй фрейм», «третий фрейм», которые ссылаются на соответствующие файлы – *first.html*, *second.html*, *third.html*. Но чтобы эти файлы отображались в нужном фрейме (а именно в правом), нужно в каждой ссылке указать имя этого фрейма (content) через параметр target:

```
<A href="first.html" target="content">первый фрейм</A>
```

- Тем самым, мы сообщаем браузеру, что файл *first.html* нужно отобразить во фрейме с именем «content» (это имя мы присвоили правому фрейму в главном документе). В тело меню добавьте заголовок «Меню».
- Создайте HTML-документ *greet.html* с заглавием «Приветствие», оформленным любым встроенным заголовком и текстом «Выберите пункт меню для перехода на соответствующую страничку».
- Создайте файлы *first.html*, *second.html*, *third.html*. В теле каждого файла вставьте заглавие «Первая (Вторая, Третья) страничка» и текст «Поздравляем! Вы попали на первую (вторую, третью) страничку!»
- Теперь откройте главный документ в браузере и проверьте работу меню. При выборе соответствующей ссылки в левом фрейме в правом отображается выбранный HTML-документ.

«Плавающий» фрейм (тэг <IFRAME>)

На страничке можно использовать так называемые **плавающие** фреймы. Это фрейм, вставленный в тело HTML-документа в виде отдельного окна (рис. 6.5).

Этот тип фреймов хорош тем, что его можно вставить в любой HTML документ. Даже в тот, у которого тело представлено не в виде фреймовой структуры, а в виде обычного тела <BODY>.

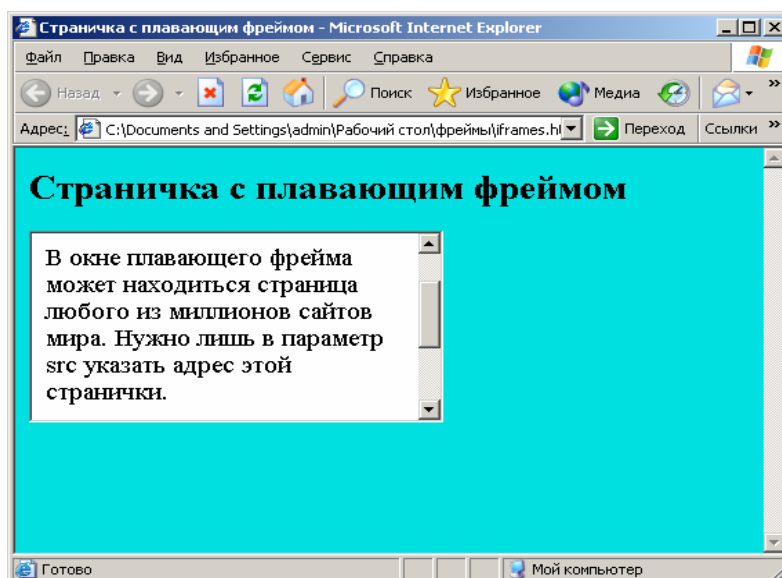


Рис. 6.5. HTML-документ с «плавающим» фреймом

«Плавающий» фрейм задается тэгом **<IFRAME>**. Например,

```
<IFRAME src=formatting.html>
```

Тэг **<IFRAME>** имеет параметры, свойственные тэгу **<FRAME>** (**src**, **frameborder**, **marginheight**, **marginwidth**, **name** и **scrolling**), и параметры, свойственные тэгу ****¹ (**align**, **width**, **height**).

Упражнение 6.4

Создайте HTML-документ *iframe.html* с плавающим фреймом (см. рис. 6.5). Страничка плавающего фрейма должна содержать текст:

Пример

В окне плавающего фрейма может находиться страница любого из миллионов сайтов мира. Нужно лишь в параметр **src** указать адрес этой странички.

Это может быть очень полезным, если вы хотите показать посетителю какую-то страницу, не переходя на нее.

<NOFRAMES></NOFRAMES>

Данный тэг используется в случае, если браузер не поддерживает фреймы. Тэг **<NOFRAMES></NOFRAMES>** размещается после тэга **<FRAMESET></FRAMESET>**. Между тэгами **<NOFRAMES>** и **</NOFRAMES>** можно коротко изложить содержание документа или перенаправить пользователя на альтернативный документ, не содержащий фреймов.

Создание более сложной структуры фреймов

При создании более сложной структуры фреймов используется вложенность конструкций **<FRAMESET></FRAMESET>** одной в другую.

Вот несколько примеров создания фреймов более сложной структуры:

<table> <tr> <td colspan="2">*</td></tr> <tr> <td>45%</td><td>55%</td></tr> </table>	*		45%	55%	<pre><FRAMESET rows="*,60"> <FRAME SRC="homepage.htm" NAME="Frame1"> <FRAMESET cols="45%,55%"> <FRAME SRC="menu.htm" NAME="Frame2"> <FRAME SRC="menu2.htm" NAME="Frame3"> </FRAMESET> </FRAMESET></pre>
*					
45%	55%				
<table> <tr> <td rowspan="2">*</td><td>15%</td></tr> <tr> <td>85%</td></tr> </table>	*	15%	85%	<pre><FRAMESET cols="*,55%"> <FRAME SRC="homepage.htm" NAME="Frame1"> <FRAMESET rows="15%,*"> <FRAME SRC="menu.htm" NAME="Frame2"> <FRAME SRC="menu2.htm" NAME="Frame3"> </FRAMESET> </FRAMESET></pre>	
*		15%			
	85%				

¹ Подробнее о параметрах тэга **** читайте в гл. 3 «Графика в HTML»

50%	50%
50%	50%

```

<FRAMESET cols="50%,50%">
  <FRAMESET rows="50%,50%">
    <FRAME SRC="homepage.htm" NAME="Frame1">
    <FRAME SRC="homepage2.htm" NAME="Frame2">
  </FRAMESET>
  <FRAMESET rows="50%,50%">
    <FRAME SRC="menu.htm" NAME="Frame3">
    <FRAME SRC="menu2.htm" NAME="Frame4">
  </FRAMESET>
</FRAMESET>

```

Задание 6.6

Создайте файл с предложенной фреймовой структурой:

Первый Frame	Второй Frame	Третий Frame
	Четвёртый Frame	

Фреймы должны содержать информацию разного характера (картинка, бегущая строка, списки и т. д.). У фреймов используйте рамки различных цветов и толщины, поля (отступы), полосы прокрутки.

Вопросы для повторения

1. Какая информация содержится в главном HTML-документе?
2. Какие тэги задают фрейм?
3. Что такое <IFRAME>?
4. Какие возможности дает использование фреймов?
5. Почему нежелательно использование фреймов в создании web-страничек?

Контрольное задание

Создайте страничку, рассказывающую о том, чему вы уже научились на занятиях по HTML (см. рис. 6.6). Используйте горизонтальное расположение фреймов. Меню разместите в верхнем фрейме, к ссылкам используйте подсказки. В нижнем фрейме должен отображаться выбранный пункт меню: отформатированный текст, списки, бегущая строка, картинка, таблица, метки внутри документа.

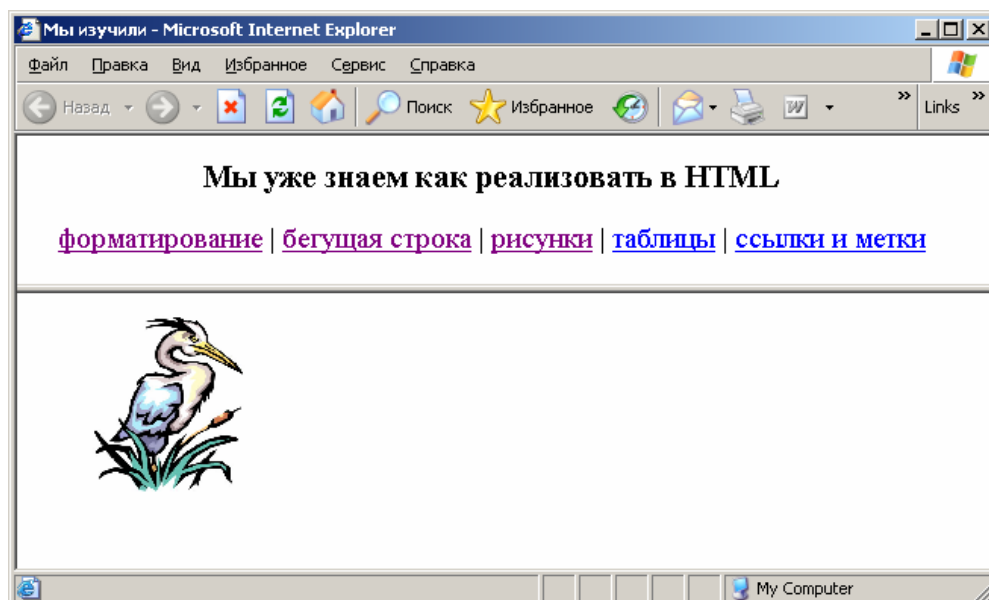


Рис. 6.6. Меню в горизонтальном фрейме

Глава 7. Создание форм

Формы – это объекты в HTML-документе, назначением которых является сбор информации от посетителей странички. С их помощью посетитель может отправить комментарии по поводу посещения сайта, посылать на сервер различные запросы или регистрироваться.

После того как пользователь заполнит форму и нажмет кнопку, внесенные им данные посылаются на сервер и обрабатываются специальной программой CGI (*Common Gateway Interface*) или PHP (*Personal Hypertext Processor*). Таким образом, пользователь может интерактивно взаимодействовать с web-сервером через Интернет.

В одном документе может быть определено несколько форм для заполнения, но форма не должна содержать в себе другую форму.

Создание формы

Процесс создания формы состоит из двух этапов. Первый заключается в создании самой формы, а второй – в создании на сервере программы, обрабатывающей данные этой формы.

Все формы создаются парными тэгами **<FORM>** и **</FORM>**, между которыми помещаются элементы формы:

```
<FORM method="get/post" action="URL программы">  
    Элементы формы  
</FORM>
```

В параметре **action** указывается адрес (*URL*) программы, которая будет обрабатывать данные формы на сервере.

Параметр **method** определяет, каким путем данные будут посылаться на сервер:

- **get**: информация из формы добавляется в конец адреса, который был указан в параметре **action**. Опытные HTML-редакторы не рекомендуют использовать метод **get**.
- **post**: при использовании данного метода информация из формы пересылается не как часть адреса, а как содержимое запроса. Данный метод рекомендуется к использованию.

Примечание. Изучение программ обработки данных формы на сервере не входит в рамки данного курса. В примерах будут рассмотрены основы создания интерфейса для отправки данных – то есть структура и основные элементы форм. Поэтому параметры *method* и *action* тэга **<FORM>** при выполнении упражнений и заданий данной главы могут быть опущены.

Для задания элементов внутри формы используются тэги **<INPUT>**, **<TEXTAREA>** и **<SELECT>**.

Рассмотрим каждый из них подробнее.

Элемент <INPUT>

Тэг <INPUT> используется для ввода текстовой информации, пересылки файлов или управляющей информации.

Параметры тэга <INPUT>:

- **name** - имя поля. Данное имя используется как уникальный идентификатор поля, по которому впоследствии программа обработки сможет получить данные, помещенные пользователем в это поле;
- **size** - определяет визуальный размер поля ввода на экране в символах (по умолчанию size=20);
- **maxlength** - определяет количество символов, которое пользователи могут ввести в поле ввода. При превышении количества допустимых символов браузер не позволяет ввести последующий символ. По умолчанию значение maxlength равно бесконечности;
- **value** – определяет начальное (по умолчанию) значение поля;
- **type** - определяет тип поля ввода. Может принимать значения:
 - **text** - описывает однострочное поле ввода .
Этот атрибут используется по умолчанию (см. пример в упр. 7.1);
 - **password** - отличается от атрибута **text** тем, что вводимое пользователем значение не отображается браузером на экране ;
 - **checkbox** (флажок) - используется, когда необходимо ответить да или нет. В форме отображается в виде ☒ . Если в форме имеется несколько полей-флажков, то можно выбрать более одного поля. Каждое поле должно иметь свое имя (*name*) и значение (*value*). На сервере обрабатываются данные только у выбранных полей;
 - **radio** - при использовании нескольких полей данного типа в одной форме пользователь может выбрать только одно поле. В форме отображается в виде ☐ . Поля должны иметь одинаковое имя (*name*), но разные значения (*value*). На сервер будет передано значение только выбранного поля;
 - **reset** - данный тип обозначает кнопку, при нажатии которой все поля формы примут значения, описанные для них по умолчанию. Атрибут *value* может содержать текст, который будет высвечен на кнопке;
 - **submit** - данный тип обозначает кнопку, при нажатии которой на сервере будет вызвана управляющая программа, описанная в параметре формы *action*. Атрибут *value* может содержать текст, который будет высвечен на кнопке;
 - **image** - данный тип обозначает кнопку *submit* в виде рисунка. Путь к картинке указывается через атрибут *src*, а синтаксис совпадает с тэгом ;
 - **file** (файловое поле) - поле передачи файла на сервер.

Упражнение 7.1

Создайте HTML-документ *simpleform.html*. Разместите в нем простую форму, предлагающую пользователю ввести свои личные данные (рис. 7.1).

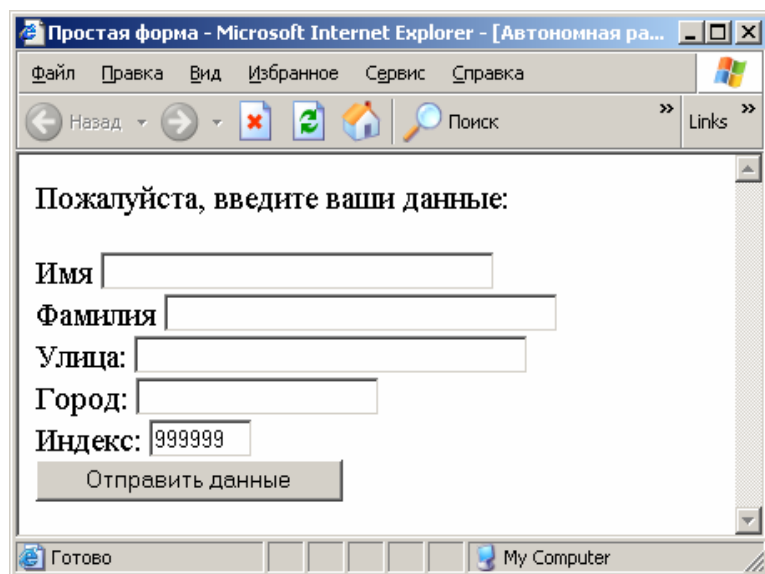


Рис. 7.1. Пример простейшей формы

Для этого:

- С помощью тэгов `<FORM>` и `</FORM>` создадим саму форму.

- Для ввода информации будем использовать элементы `<INPUT>`, которые будут представлять собой текстовые поля (параметр `type=text`). В эти поля пользователь будет вносить свои данные, которые будут передаваться на сервер как значения переменных с соответствующими именами (параметр `name`).

- В текстовом поле Индекс зададим ограничение на количество символов (параметр `maxlength`) и значение по умолчанию (параметр `value`):

```
<INPUT name="zip" size=6 maxlength=6 value="999999">
```

- В конце формы поместим кнопку, при нажатии на которую данные будут отсылаться на сервер (тэг `<INPUT>` с параметром `type=submit`).

```
<INPUT type=submit value="Отправить данные">
```

- В результате мы получим такую структуру формы:

```
<FORM>
<P>Пожалуйста, введите ваши данные:</P>
<P>Имя <INPUT name="name" type=text size=35> <BR>
Фамилия <INPUT name="surname" type=text size=35> <BR>
Улица: <INPUT name="street" type= text size=35> <BR>
Город: <INPUT name="city" type= text name=20 maxlength=20> <BR>
Индекс: <INPUT name="zip" size=6 maxlength=6 value="999999"> <BR>
<INPUT type=submit value="Отправить данные"></P>
</FORM>
```

Задание 7.1

Выровняйте элементы формы, созданной в упр. 7.1 путем размещения их в ячейках таблицы (рис. 7.2).

Рис. 7.2. Элементы формы выровнены с помощью ячеек таблицы

Упражнение 7.2

Создайте HTML-документ *anketa1.html* с формой-анкетой, предлагающей выбрать любимое блюдо (с возможностью выбора нескольких пунктов).

Для этого:

- Для описания пунктов выбора используйте элемент `<INPUT>` с параметром `type=checkbox` (при просмотре в окне браузера рядом с пунктом появится квадратик):

```
...
<P>Выберите ваше любимое блюдо:
<FORM>
<INPUT type="checkbox" name="food1" value="Салат Оливье">Салат Оливье<BR>
<INPUT type="checkbox" name="food2" value="Мясо по-испански"> Мясо по-испански<BR>
<INPUT type="checkbox" name="food3" value="Макароны по-флотски" checked>Макароны по-флотски<BR>
<INPUT type="submit" value="Отправить данные">
</FORM>
...
```

Рис. 7.3. Анкета с возможностью выбора нескольких вариантов

- Посмотрите результат в окне браузера.
- Проанализируем работу формы: при нажатии на кнопку «Отправить данные» на сервер будут отосланы одна, две или три переменных (в зависимости от количества выбранных вариантов) с именами, хранящимися в

параметре name (food1, food2, food3) и с соответствующими значениями, хранящимися в параметре value («Салат Оливье», «Мясо по-испански», «Макароны по-флотски»).

- Обратите внимание на то, что элемент «Макароны по-флотски» указан как заранее отмеченный.

Задание 7.2

Создайте HTML-документ *anketa2.html* с формой-анкетой, предлагающей выбрать самый любимый напиток с возможностью выбора только одного варианта (рис. 7.4).

Примечание. Проанализируйте тот факт, что на сервер нужно отправить только одно значение, следовательно, на сервер будет отсылаться одна и та же переменная (параметр name одинаковый у всех элементов), но с разными значениями (параметр value), в зависимости от выбранного пункта.

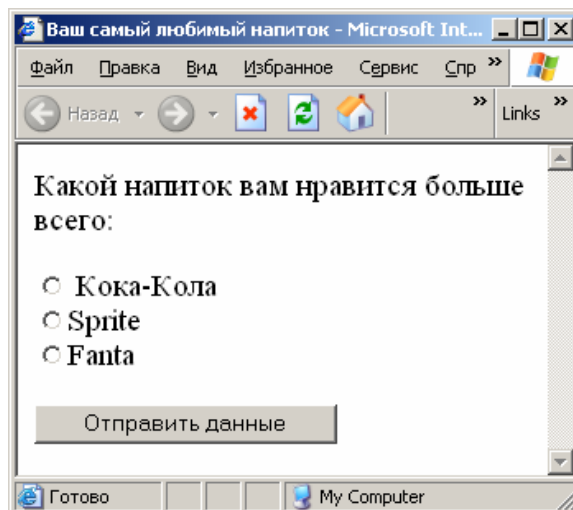


Рис. 7.4. Анкета с возможностью выбора только одного варианта

Упражнение 7.3

Создайте HTML-документ *password.html*, предлагающий ввести пароль (рис. 7.5).

Для этого:

- У элемента `<INPUT>` значение параметра задайте password.
- Загрузите форму в окне браузера. Введите в поле пароль — в поле отображаются не символы, а точки.

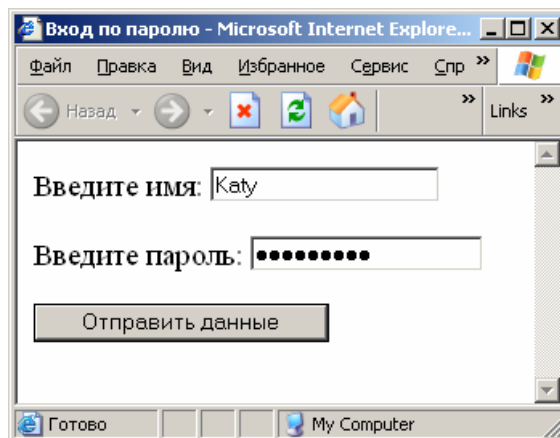


Рис. 7.5. Окно с формой для ввода пароля

Упражнение 7.4

В HTML-документе *simpleform.html*, созданном в упр. 7.1, в форму добавьте кнопку сброса значений полей формы.

Для этого:

- Добавьте новый элемент `<INPUT>` параметром `type=reset`:

`<INPUT type=reset value="Очистить поля">`

- Проверьте работы кнопки — при ее нажатии все поля очищаются или заполняются значением, заданным по умолчанию.

Упражнение 7.5

Создайте документ *file.html* с формой, позволяющей отправлять на сервер файлы.

Для этого:

- Создайте HTML-документ с текстом «Выберите файл для загрузки на сервер».
- Создайте в этом документе форму.
- Добавьте в нее элемент `<INPUT>`, позволяющий выбрать файл (`type=file`)

`<INPUT type=file name="Отправка файла">`

и кнопку «Отправить».

Примечание. Для отправки файла в параметре *action* тэга `<FORM>` вводится адрес специальной программы, осуществляющей пересылку файла на сервер (в примере адрес указывать не надо).

Элемент `<TEXTAREA>`

Тэг `<TEXTAREA>` используется для ввода большого количества текстовой информации (в несколько строк).

Атрибуты тэга `<TEXTAREA>`:

- **name** - имя поля;
- **cols** - число колонок (символов) в текстовой области;
- **rows** - число видимых строк в текстовой области.

Если вы хотите, чтобы в поле ввода по умолчанию выдавался какой-либо текст, то необходимо вставить его внутри тэгов `<TEXTAREA>` и `</TEXTAREA>`.

Упражнение 7.6

Создайте документ *otziv.html* с формой, предлагающей написать отзыв о статье (рис. 7.6).

Для этого:

- Создайте HTML-документ с текстом «Напишите отзыв о прочитанной статье».
- В HTML-документе создайте форму.
- Добавьте в нее элемент `<TEXTAREA>` с 6 строками и шириной 64 символа, в котором будет выведена подсказка «Здесь введите свой отзыв»:

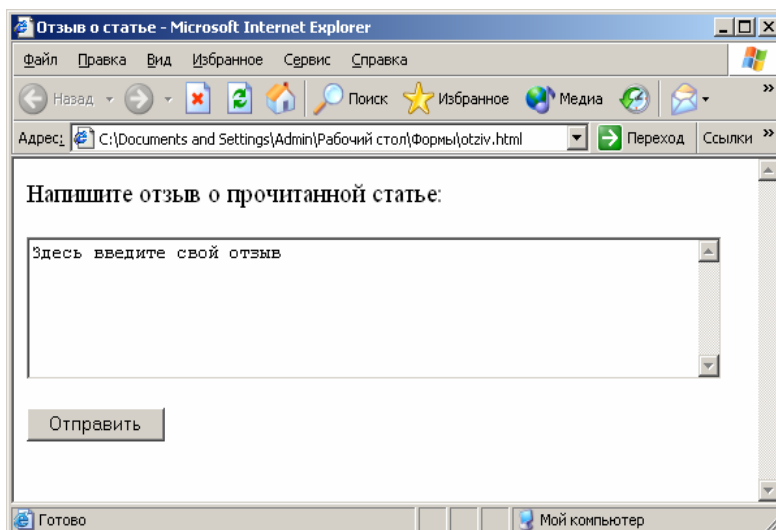


Рис. 7.6. Использование элемента `<TEXTAREA>`

```
<TEXTAREA NAME="otziv" COLS=64 ROWS=6>Здесь введите свой  
отзыв</TEXTAREA>
```

- В конце формы добавьте кнопку «Отправить».

Элемент <SELECT>

Элемент `<SELECT>` дает возможность пользователю выбрать значение из фиксированного списка значений. Обычно этот элемент представлен выпадающим списком.

Для определения списка пунктов используются тэги `<OPTION>` между `<SELECT>` и `</SELECT>`.

```
<SELECT ....>
  <OPTION>...
  <OPTION>...
  ...
</SELECT>
```

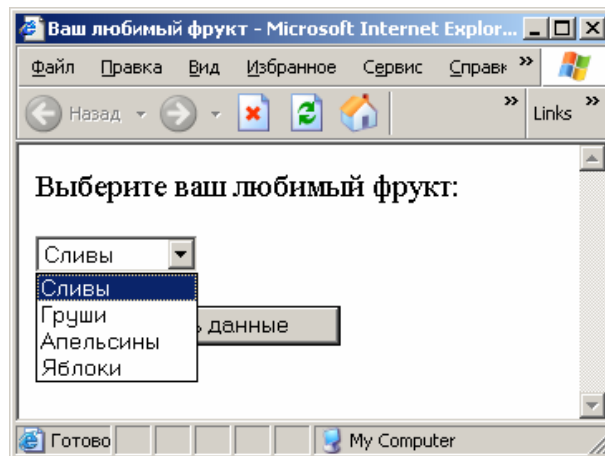


Рис. 7.7. Элемент `<SELECT>` в виде выпадающего списка

Параметры тэга `<SELECT>`:

- **name** - имя объекта `SELECT`;
- **multiple** - позволяет выбрать более чем одно наименование;
- **single** – из списка можно выбрать только один вариант (этот параметр задан по умолчанию);
- **size** - определяет число пунктов, видимых для пользователя. Если этот параметр не указан, то список на экране отображается в виде выпадающего списка (видно одно наименование), если `size>1`, то браузер представляет на экране обычный список (число - количество видимых наименований).

Параметры тэга `<OPTION>`:

- **selected** – указывает, какой пункт выбран по умолчанию;
- **value** – значение, которое передается на сервер для обработки. По умолчанию значение `value` равно элементу `<OPTION>`.

Упражнение 7.8

Создайте документ *fruit.html*. Поместите в него форму, предлагающую пользователю выбрать любимый фрукт из выпадающего списка (рис. 7.7).

Для этого:

- Для создания списка используйте элемент `<SELECT>`.
- Обязательно задайте у тэга `<SELECT>` параметр `name` – в переменной с этим именем будут переданы данные на сервер.
- Каждый элемент списка опишите с помощью тэга `<OPTION>`.
- Не забудьте закрыть элемент `<SELECT>` тэгом `</SELECT>`.
- Добавьте в форму кнопку «Отправить данные».

Задание 7.3

В форме, созданной в упр. 7.8, измените элемент <SELECT> таким образом, чтобы отображался полный список фруктов. Установите значение, выбранное из списка по умолчанию – «Яблоки» (рис. 7.8).

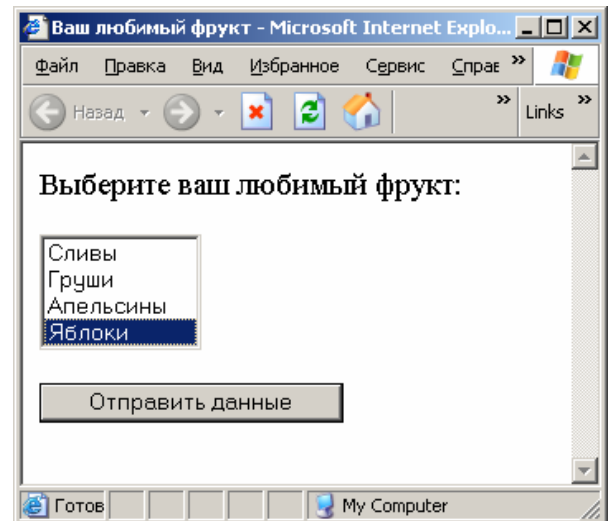


Рис. 7.8. Простой список

Вопросы для повторения

1. Для чего при создании web-страничек используют формы?
2. С помощью какого тэга задают форму?
3. Какие элементы могут находиться на форме?

Контрольное задание

Создайте форму по образцу:

Анкета пользователя

Ответьте, пожалуйста, на вопросы:

Ваша фамилия:

Ваше имя:

Ваше отчество:

Ваш пол:

☐ Мужчины

☐ Женщины

Семейное положение:

☐ Женат/замужем

☐ Холост/Не замужем

Количество детей

Место жительства:

Страна:

Другая

Индекс:

Город:

Улица:

Дом: Квартира:

Электронный адрес:

Глава 8. Каскадные таблицы стилей

Используя тот или иной тэг из стандартного набора, мы задаем стандартный (определенный по умолчанию) внешний вид абзацу, символам, рисунку и т. д. Если же мы хотим получить иной вид, то нужно использовать дополнительные тэги.

Например, чтобы создать абзац с полужирным начертанием, красного цвета и с подчеркиванием, нужно использовать 3 тэга - , и <U>:

```
<P><font color="red"><B><U>Абзац с красным полужирным подчеркнутым шрифтом  
</U></B></FONT></P>
```

Если подобный стиль нужно использовать не один, не два, а десятки раз, то использование такого набора тэгов становится нерациональным и неудобным для разработчика.

Решается такая проблема с помощью CSS (*Cascading Style Sheets* - каскадные таблицы стилей). CSS позволяют задать шаблон стиля и применять его в дальнейшем.

Использование каскадных таблиц стилей имеют ряд преимуществ:

1. CSS предоставляет гораздо больший объем параметров форматирования объектов HTML¹.
2. Используя CSS, зачастую можно легко сократить размер исходного HTML-документа.
3. CSS позволяет «настраивать» такие объекты HTML, как таблицы, списки, гиперссылки и т. д.

Первые каскадные таблицы стилей CSS были реализованы в браузере Internet Explorer 3.0. С момента своего возникновения структура CSS была несколько раз пересмотрена, в нее были добавлены новые элементы и убраны старые.

Способы задания CSS

Существует три способа задания стилей, применение каждого из которых обусловлено конкретной задачей.

1-й способ - строчный (внутренние таблицы стилей - *Inline Style Sheets*) - стили вставляются прямо в описание тэгов HTML с помощью специального параметра **style**.

<ТЭГ style="описание стиля">текст</ТЭГ>

Этот способ обычно используется, если стиль нужно применить единожды к конкретному тэгу и требуется использование параметров, доступных только через CSS.

¹ Подробный перечень параметров, доступных при использовании CSS, смотрите в Приложении III в разделе «Приложения» данного пособия.

Упражнение 8.1

Создайте HTML-документ *inline.html*, содержащий абзац с разреженным шрифтом и таблицу с жирной верхней границей (см. рис. 8.1).

Для этого:

- Во-первых, без использования CSS здесь не обойтись, потому что разреженный интервал невозможно создать стандартными средствами HTML, следовательно, прибегнем к использованию CSS. Создаем стиль у тэга `<P>`:

```
<P style="letter-spacing:12">Разреженный текст в абзаце задан с помощью CSS</P>
```

- Аналогично создаем стиль у тэга `<TABLE>` (параметры, доступные без CSS описывает вне «style»):

```
<TABLE style="border-top-width:10px; border-color:blue;" border=1 cellpadding=0 cellspacing=0 bordercolor=blue >
```

- Во-вторых, будем использовать строчный способ задания CSS, так как стиль у каждого элемента (абзац и таблица) нужно применить единожды.

В данном примере описаны два стиля: стиль абзаца и стиль таблицы. При описании каждого стиля использовались стандартные и новые атрибуты.

Например, `letter-spacing` задает разреженность текста, `border-top-width` - толщину верхней границы, а `border-color` – цвет границы таблицы.

Примечание. Если в описании стиля используются несколько атрибутов, то они должны быть перечислены через «;».

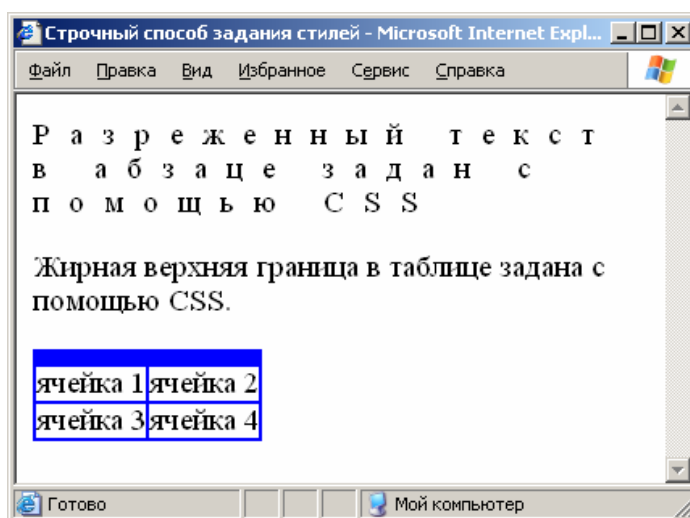


Рис. 8.1. Использование строчного способа задания стилей

2-й способ - вложенный (глобальные таблицы стилей - *Global Style Sheets*): стили описываются все в одном месте между тэгами `<STYLE>` и `</STYLE>` (обычно этот раздел размещается в разделе `<HEAD>` `</HEAD>`).

```
<STYLE>
  тэг1 {описание стиля}
  тэг2 {описание стиля}
  ...
  тэг{описание стиля}
</STYLE>
```

Используя этот способ, мы переназначаем форматирование тэгов. Этот способ обычно применяется, если стиль тэга в документе используется

несколько раз. Тогда, описав стиль один раз в разделе <STYLE>, при указании тэга в тексте уже не нужно стиль описывать повторно.

Упражнение 8.2

Создайте HTML-документ *seti.html* по образцу (рис. 8.2). Названия классификаций заданы тэгом <P>, перечисления – элементы маркированного списка (тэг).

Элементы имеют следующие параметры форматирования:

Тэг <P>	Межсимвольный интервал - 10 пикселей, начертание - полужирное, размер шрифта – 20 пикселей
Тэг 	Начертание – нежирный курсив, размер шрифта – 18 пикселей

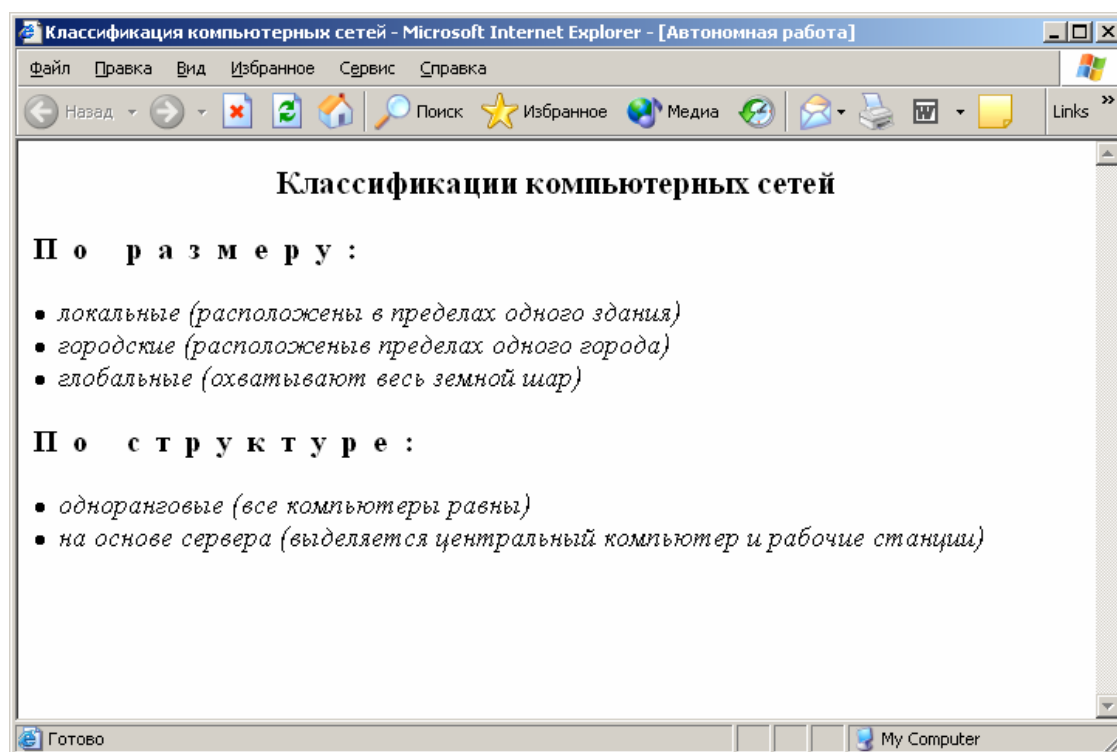


Рис. 8.2. Использование строчного способа задания стилей

Для этого:

- Так как в числе описанных параметров встречаются нестандартные, то опять же прибегнем к использованию CSS.
- При описании стилей будем использовать вложенный способ, так как в тексте примера один и тот же стиль будет применяться несколько раз.
- Описываем стили в разделе <STYLE> </STYLE>:

```
P {letter-spacing:10px;font-style:normal;font-weight:bold;font-size:20px;}
LI {font-style:italic; font-size:18px;font-weight:none;}
```

- Теперь создаем текст HTML-документа, указывая нужные тэги (стили описывать не нужно).

3-й способ - внешний (связанные таблицы стилей - *Linked Style Sheets*): отличается от предыдущего тем, что описания стилей помещаются не в разделе

<STYLE>, а вносятся в отдельный тестовый файл с расширением *.css (например, style.css). Если мы хотим использовать описанные стили на страничке, мы должны включить ссылку на этот файл в наш HTML-документ между тэгами <HEAD> и </HEAD> с помощью тэга <LINK>.

<LINK rel=stylesheet type=text/css href="имя файла с описанными стилями">

Этот способ применяется, если описанные стили используются не в одном документе, а сразу в нескольких – мы просто вставляем ссылку на файл с описаниями стилей в каждый HTML-документ.

Упражнение 8.3

Создайте HTML-документ *inform.html* (рис. 8.3). Оба HTML-документа – *seti.html* и *inform.html* должны использовать одни и те же параметры форматирования, описанные в упр. 8.2.

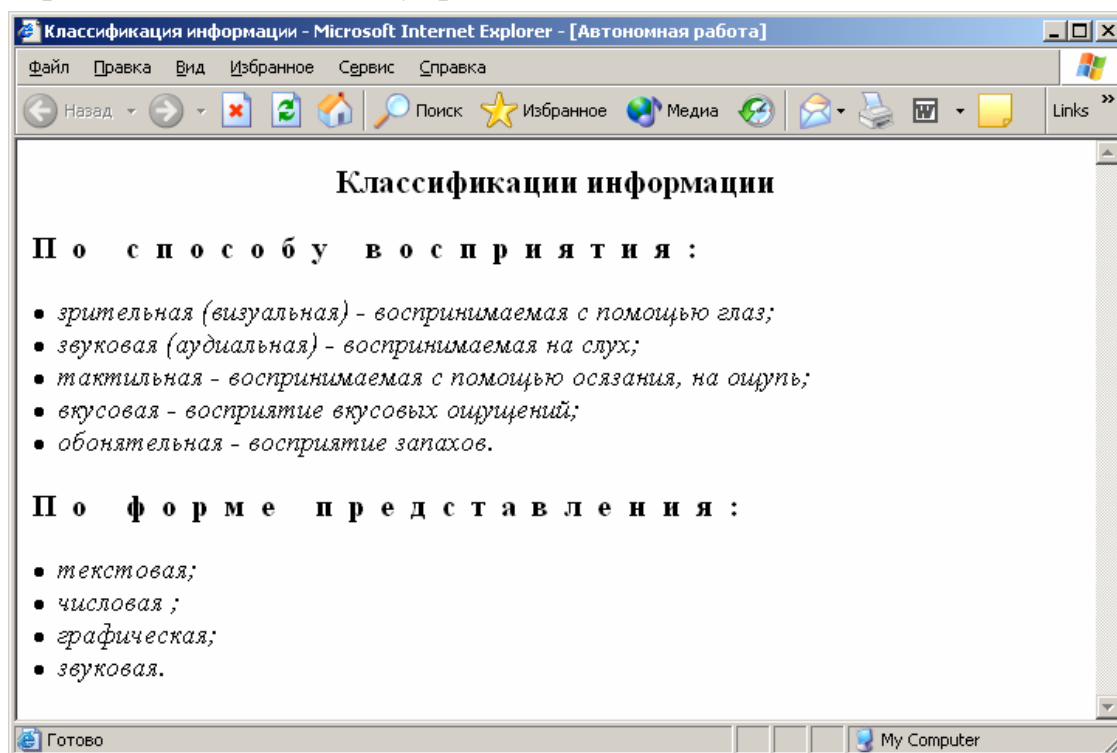


Рис. 8.3. HTML-документ *inform.html*, использующий те же параметры форматирования, что и *seti.html*

Для этого:

- Будем использовать внешний способ задания стилей, так как у нас имеются две странички, использующие один и тот же набор параметров форматирования.

- Создайте файл *style.css*.

- Переместите описание стилей из HTML-документа *seti.html* в css-файл. Сохраните изменения.

- В обоих HTML-документах в разделе <HEAD> </HEAD> добавьте ссылку на созданный css-файл с описанными стилями.

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

Задание 8.1

Создайте HTML-документ по образцу (рис. 8.4), используя наиболее подходящий способ задания стиля:

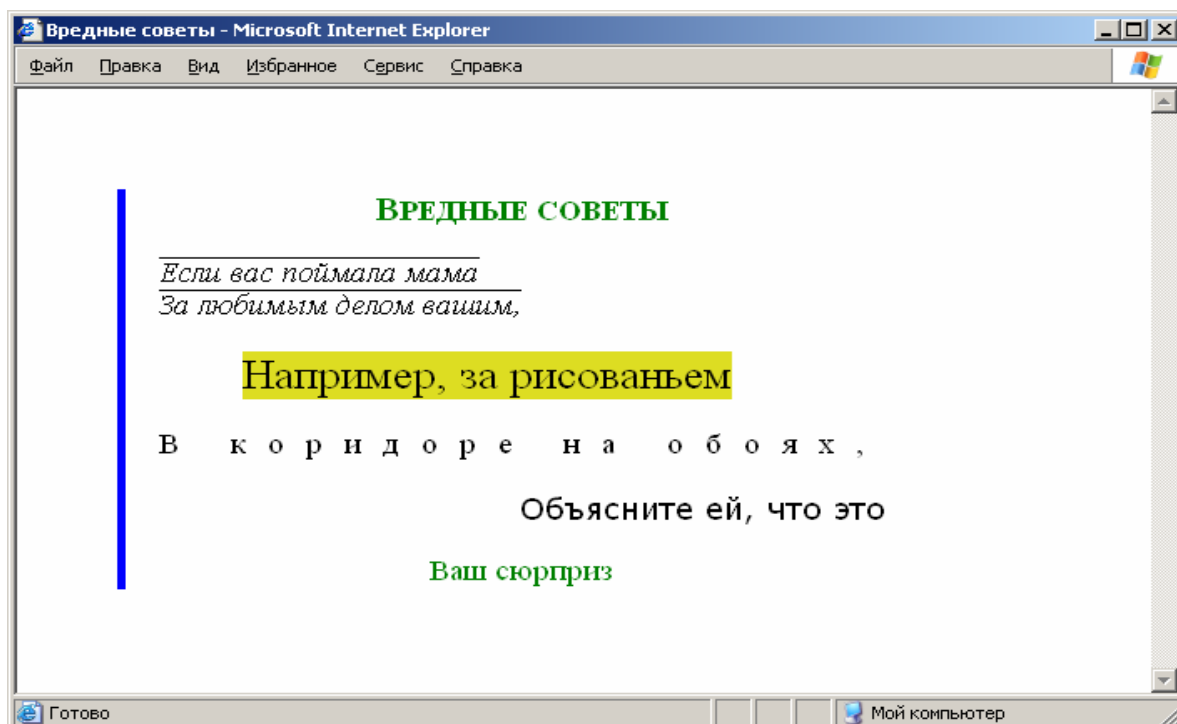


Рис. 8.4. Образец выполненного задания

Примечание. Для задания границы слева используйте таблицу. Цвета выберите на ваше усмотрение.

Задание 8.2

Создайте HTML-документ по образцу (см. рис. 8.5), используя подходящий способ задания стилей.

Примечание. При выполнении упражнения используйте следующие параметры форматирования:

| Стиль | Параметры форматирования |
|------------------------------------|--|
| Заголовок | Выравнивание: по центру; Courier New, полужирный, 18 пт, цвет красный |
| Подзаголовок | Нумерованный список, цвет шрифта – зеленый, межсимвольный интервал – разреженный на 5пт, начертание – жирный |
| Текст абзацев | Шрифт Verdana, размер шрифта – 11пт |
| Выделяемые слова
внутри абзацев | Подчеркнутый курсив, цвет синий |

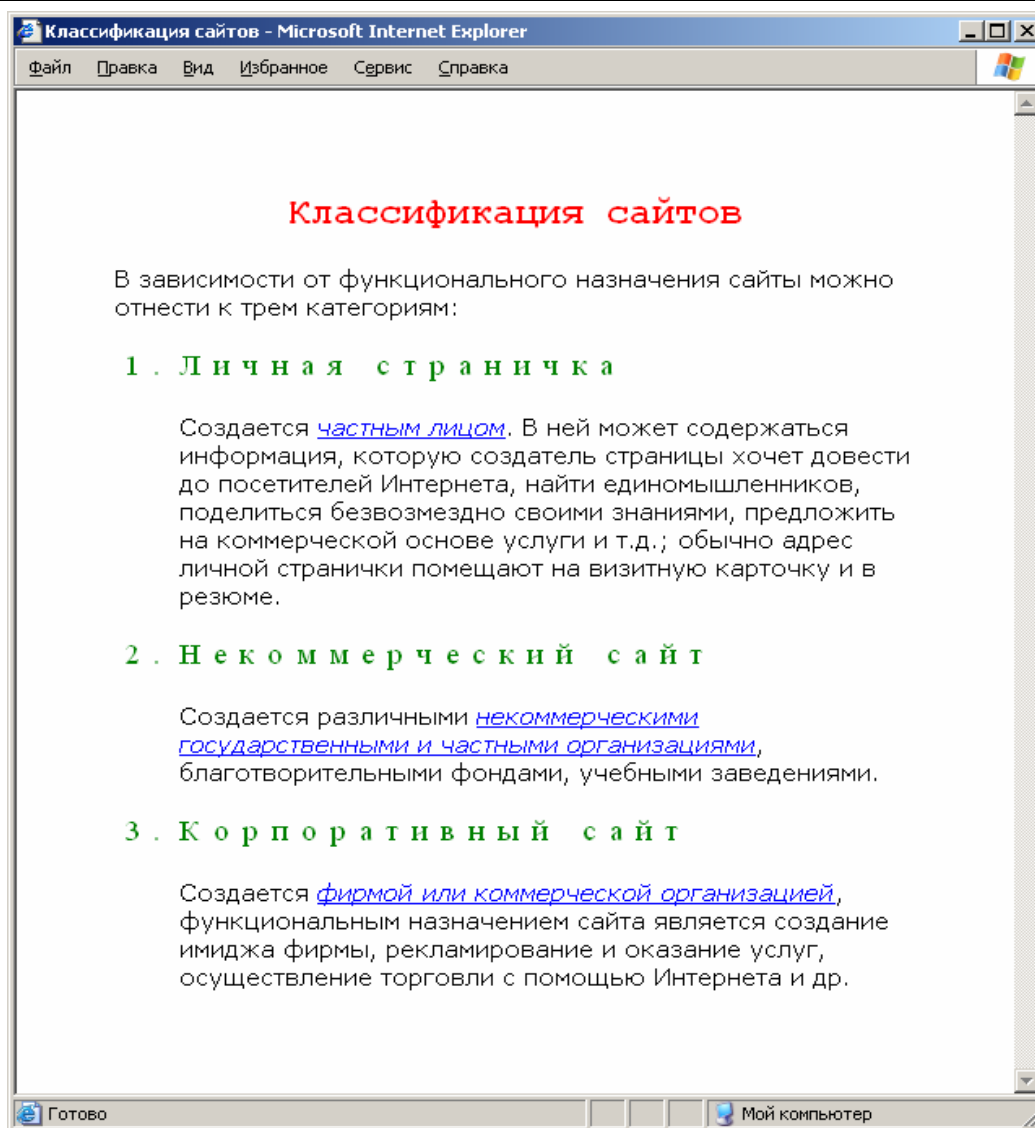


Рис. 8.5. Образец выполненного задания 8.2

Задание 8.3

Создайте еще один HTML-документ, отформатированный в том же стиле, что и в зад. 8.2. При форматировании используйте подходящий способ задания стилей.

Классы

Итак, используя шаблоны стилей, можно настраивать тэги по своему усмотрению. Благодаря стилям тэг приобретает новые параметры форматирования.

Но иногда требуется использование нескольких вариантов отображения одного и того же элемента в HTML-документе. Например, несколько абзацев (тэг <P>) имеют вид, отличающийся от других.

Спецификация HTML позволяет использовать для одного и того же элемента несколько стилевых правил, интерпретируемых браузером последовательно, каскадом.

Для этого используются **классы стилей** (вложенные каскады). Каждое новое отображение тэга представляется новым классом с уникальным именем, которое начинается с точки:

```
P.zagolovok {text-align: center; font-weight: bold; font-size:16px;},
```

В данном примере задан класс `zagolovok`, который описывает набор параметров форматирования тэга `<P>`: выравнивание, полужирное начертание, размер шрифта 16 пх.

Если задается несколько классов одного тэга, принадлежность класса определенному тэгу определяется в первом описании, а в последующих описаниях тэга можно не указывать. Например,

```
P {text-align: center; font-weight: bold; font-size:16px;}  
  .text {text-align: left; font-weight: normal; font-size: 12px;}  
  .right {text-align:right}
```

Способы задания классов

Существует два способа задания классов:

1. **Внутренний** – в разделе `<STYLE></STYLE>`.
2. **Внешний** - в отдельном файле с расширением `.css`.

Применение класса в HTML-документе осуществляется с помощью параметра `class`. Например,

```
<P class=zagolovok>
```

Можно задать универсальный класс – класс, который можно применять не к одному, а к нескольким тэгам. Этот класс описывается самым первым или после описания классов тэга `<BODY>`, и при его задании тэг не указывается. Например,

```
.left {text-align: left}
```

Созданный в примере класс `left` можно будет применить к тэгу `<P>`, `<TD>`, `<Hx>` и другим тэгам, где этот параметр форматирования применим.

Упражнение 8.4

Работа по созданию и применению классов.

Создайте HTML-документ `class.html` (см. рис. 8.6). Для форматирования текста создайте два класса тэга `<P>` с различными параметрами форматирования.

Для этого:

- Введите текст странички.
- В разделе `<STYLE></STYLE>` создайте два класса у тэга `<P>`: `heading` – для заголовков и `text` – для основного текста.

```
P .heading {font-family:Arial; background:yellow; font-weight:bold}  
  .text {font-family:Times New Roman; padding-left:130}
```

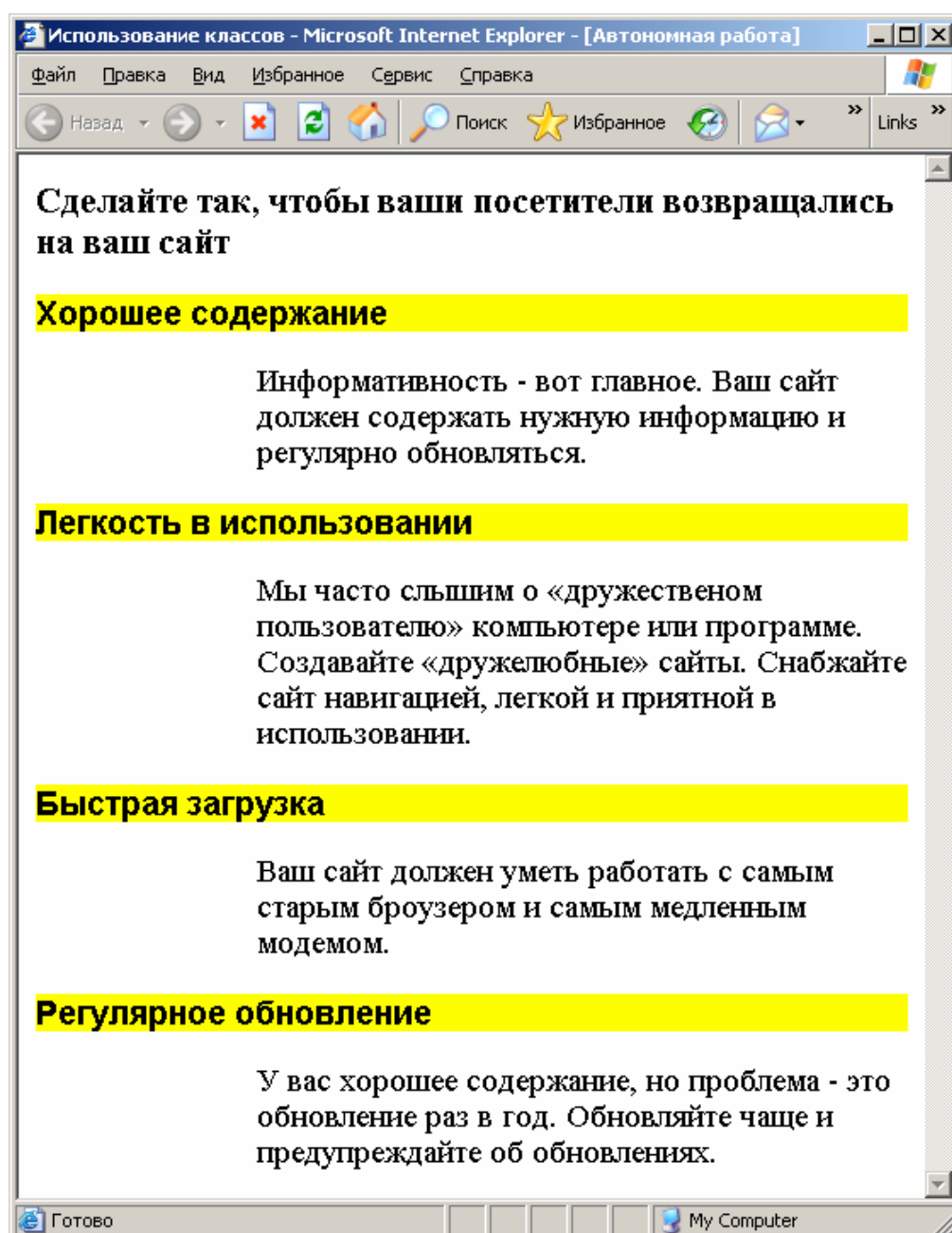


Рис. 8.6. Использование классов при форматировании текста

Задание 8.4

Создайте HTML-документ по образцу (рис. 8.7), задав классы у тэга для выделения элементов окна и пунктов меню.

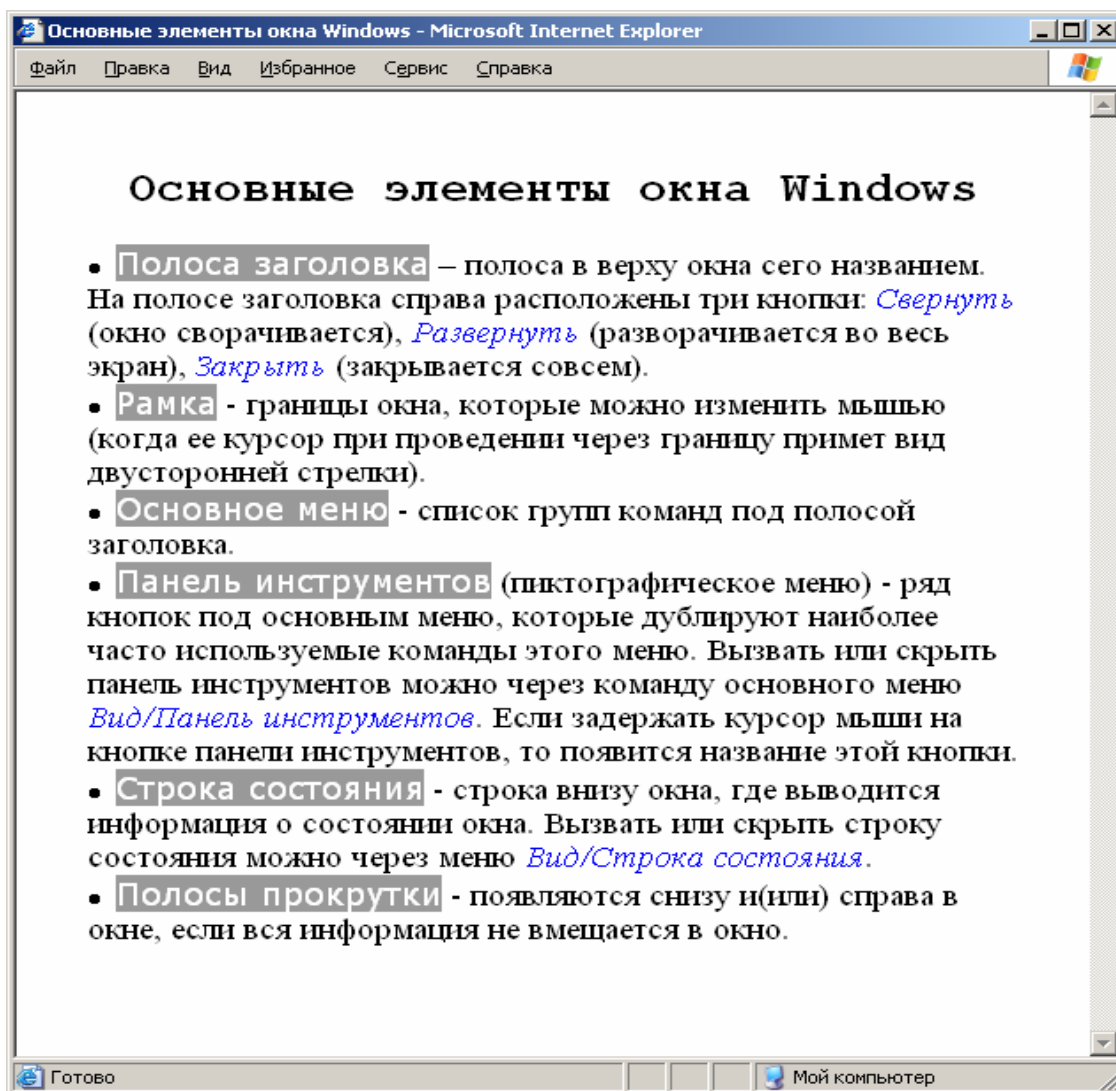


Рис. 8.7. Образец выполненного задания 8.4

Позиционирование элементов HTML

Все элементы языка разметки HTML выводятся в той последовательности, в какой размещены тэговые конструкции. С помощью средств CSS можно прибегнуть к позиционированию необходимых объектов в пределах web-странички. Позиционирование – это пространственное размещение элементов HTML-документа.

Существует два типа визуального позиционирования элементов:

- абсолютное;
- относительное.

Позиционирование элемента задается с помощью CSS через параметр `style`, где параметры перечисляются через точку с запятой:

```
<ЭЛЕМЕНТ style="параметры позиционирования">
```

В параметры позиционирования входят:

1. Параметры размещения элемента:

- **position** - устанавливает, каким образом вычисляется положение слоя на странице. Возможные значения:
 - *absolute* — координаты позиции элемента будут вычисляться от верхнего левого угла странички;
 - *relative* — координаты будут вычисляться от верхнего левого угла родительского элемента, т. е. элемента, в котором находится позиционируемый элемент;
 - *static* — элемент всегда будет находиться в зафиксированной месте окна браузера при прокручивании странички;
- **top** - устанавливает вертикальную координату элемента (в пикселях: px, в процентах: % - от ширины родительского элемента);
- **left** - устанавливает горизонтальную координату элемента (в пикселях: px, в процентах: % - относительно ширины родительского элемента);
- **width** - устанавливает ширину элемента (в пикселях: px, в процентах: % - от ширины родительского элемента);
- **height** - устанавливает высоту элемента (в пикселях: px, в процентах: % — от ширины родительского элемента).

2. Параметры «видимости» элемента:

- **visibility** — устанавливает, будет ли виден элемент. Возможные значения:
 - *visible* — слой будет виден;
 - *hidden* — слой не будет виден;
- **z-index** — указывает, в каком порядке слои будут перекрываться. Слои с большим **z-index** будут перекрывать слои с меньшим **z-index**. Возможны любые значения, включая отрицательные. В случае отрицательных значений элементы данного слоя будут закрыты обычным текстом; с положительным значением — будут лежать «перед» обычным текстом, перекрывая его. По умолчанию $z-index=0$.

Упражнение 8.5

Создайте HTML-документ *position.html*, содержащий абзац «Позиционирование в HTML» с тенью (рис. 8.8).

Для этого:

- Создайте в HTML-документе абзац с заданным текстом. Отформатируйте его: измените цвет текста, шрифт, размер шрифта.
- Для создания тени

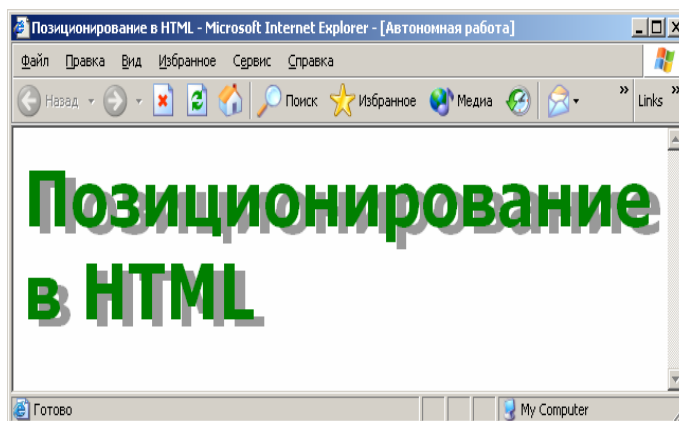


Рис. 8.8. Создание тени с помощью позиционирования

используйте копию абзаца, но со смещением. Для этого у тэга <P> через CSS задайте координаты размещения второго абзаца:

```
<P style="position:absolute;top:20px;left:20px;">
```

- Посмотрите страничку в окне браузера.
- Чтобы «тень» была на втором плане, задайте z-index у «тени» меньший, чем у основного текста (у которого z-index по умолчанию равен нулю). Например, -1.

```
<P style="position:absolute;top:20px;left:20px; z-index:-1">
```

Слои

Обычно в позиционировании используют слои.

Слой – это HTML-объект, который объединяет несколько элементов, расположенных на HTML-документе (таблицы, текст, картинки) в одну группу.

Использование слоев позволяет управлять группой элементов в целом, в том числе и позиционировать. Слои делают web-страницу компактной, динамичной, неожиданной и неотразимо привлекательной.

Слой задается тэгами <DIV> и </DIV>:

```
<DIV>элементы слоя</DIV>
```

Таким образом, чтобы сгруппировать несколько элементов в слой, нужно поместить их между тэгами <DIV> и </DIV>.

Упражнение 8.6

Создайте HTML-документ *sloi.html* (рис. 8.9). При позиционировании используйте слои.

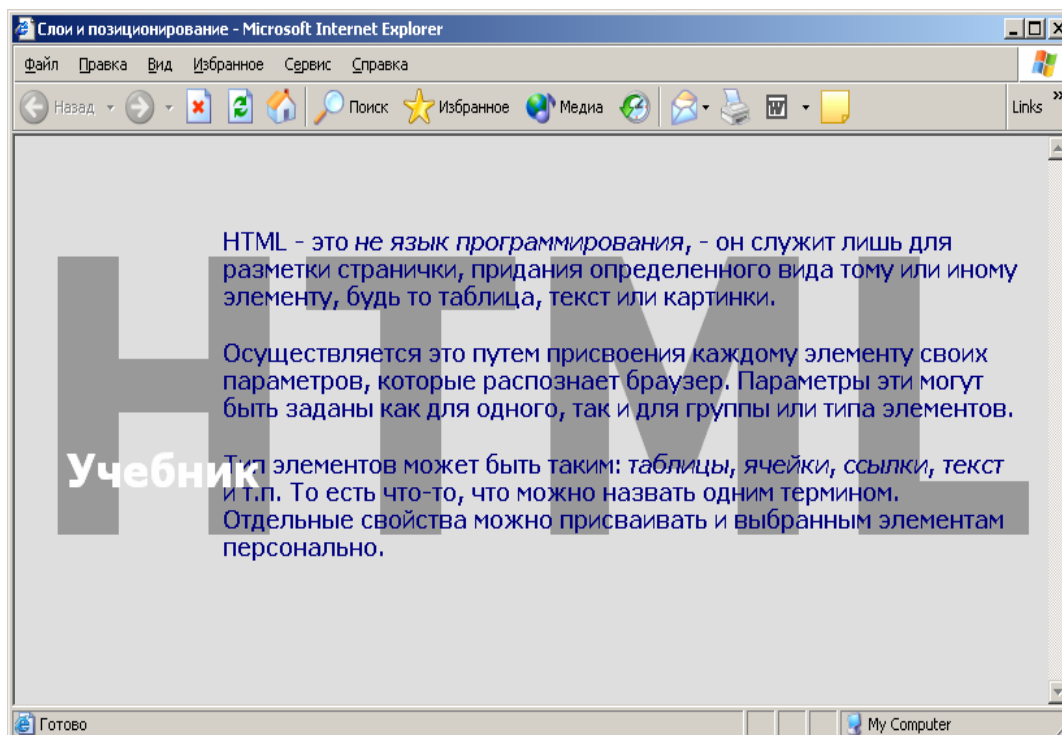


Рис. 8.9. Использование слоев в позиционировании

Для этого:

- Создайте три слоя: для слова «HTML», слова «Учебник» и основного текста.
- Разместите слои в HTML-документе, используя позиционирование через CSS (параметры размещения каждого слоя определите сами):

```
<DIV style="position:absolute; top=...; left=...">элементы слоя</DIV>
```

- Определите порядок расположения слоев, задав у каждого слоя z-index.
- Просмотрите результат в окне браузера.

Вопросы для повторения

1. Каковы преимущества использования стилей в HTML?
2. Назовите и опишите способы задания стилей, когда рекомендуется тот или иной способ?
3. Что такое классы? Когда необходимо задавать классы?
4. Назовите и опишите способы задания классов.
5. Что такое позиционирование? Возможна ли работа с позиционированием без CSS?
6. Дайте понятие слоя, для чего используются слои?

Контрольное задание

Создайте HTML-документ по образцу, используя классы, слои и позиционирование.

HTML

CSS

JavaScript

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

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

~~Учебник CSS~~

Проектное задание

Разработка сайта «Моя личная страничка»

Примерное техническое задание на разработку сайта «Моя личная страничка»

I. Описание сайта

Сайт относится к категории «личная страничка». На сайте размещена информация о владельце сайта: его биография, увлечения, фотографии и т. д.

II. Структура сайта

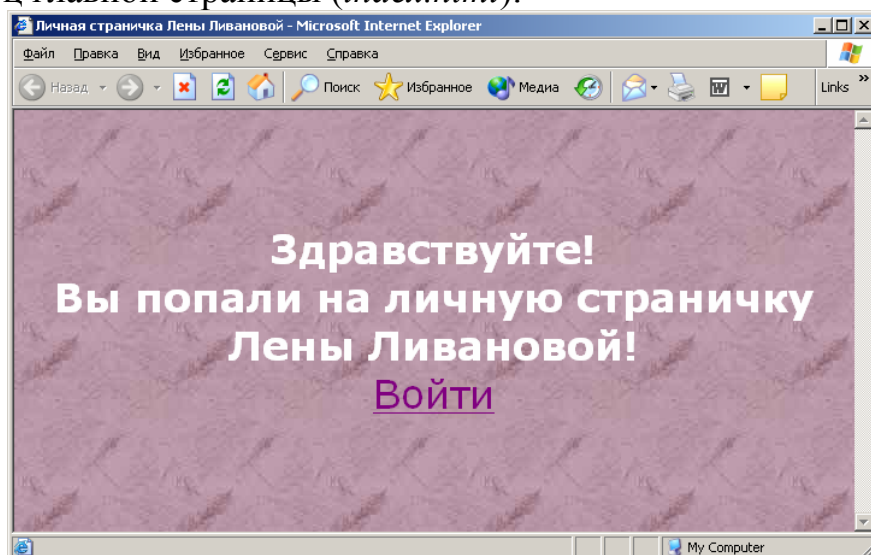
Посетитель сайта вначале попадает на главную (титულную) страницу, затем по ссылке переходит на второстепенную страницу (макеты титульной и второстепенной страниц см. п. III «Макеты страниц»). Второстепенная страница имеет фреймовую структуру, состоящую из четырех фреймов: левый фрейм – меню-список всех разделов сайта, верхний фрейм – бегущая строка, нижний фрейм – ссылка на адрес электронной почты автора странички, центральный фрейм – содержание изменяется в зависимости от выбранного раздела в меню.

Сайт включает в себя следующие разделы:

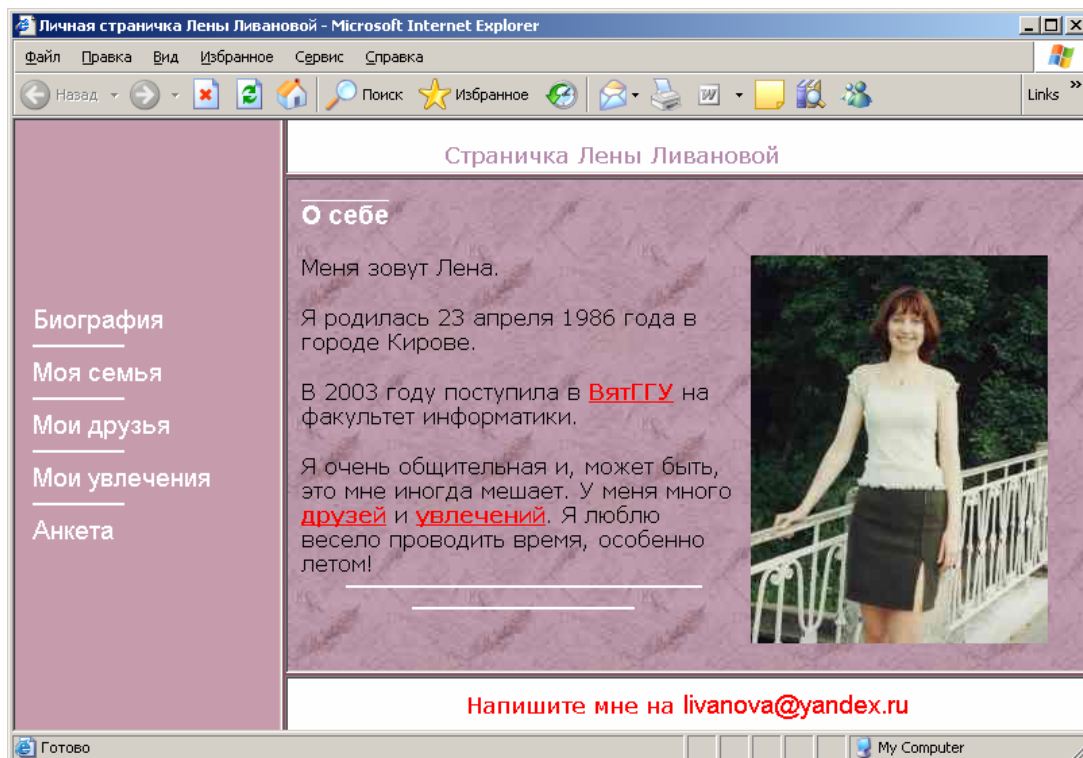
1. «Биография»: краткая биография о себе, сопровождаемая фотографиями разных лет.
2. «Моя семья»: небольшой рассказ о членах семьи, сопровождаемый фотографиями.
3. «Мои друзья»: фотографии друзей и подписи к ним.
4. «Мои увлечения»: перечисление увлечений и краткий рассказ об одном из них.
5. «Анкета»: форма, предлагающая внести личную информацию посетителя и пожелания по сайту.

III. Макеты страниц

Образец главной страницы (*index.html*):



Образец второстепенной страницы (*main.html*).



IV. Форматирование сайта

1. Страницы всех разделов оформлены в едином стиле: одинаковое форматирование основного текста, заголовков разделов, текста ссылок, используется один фоновый рисунок на страницах всех разделов.
2. Текст главной страницы сайта расположен по центру по горизонтали и по вертикали.
3. Меню выровнено по центру по вертикали.

Примечания по выполнению проектного задания

1. Используйте интересное поведение бегущей строки. Например, достигнув края фрейма, она меняет свое направление на противоположное.
2. Используйте альтернативный текст у фотографий и рисунков.
3. При перечислении увлечений используйте список.
4. Параметры форматирования, используемые на всех страничках, опишите в файле *style.css*.
5. На сайте должны быть задействованы все изученные параметры форматирования текста.
6. В разделе «Мои друзья» фотографии разместите в несколько рядов (используйте таблицу). При этом на страничке разместите только уменьшенные фотографии, которые являются ссылками на большие фотографии, открывающиеся в новом окне.

Приложения

Приложение I

Web-палитра цветов

Цвет	Имя цвета	Значение
Аквамарин	Aquamarine	#7FFFD4
Античный белый	Antiquewhite	#FAEBD7
Бежевый	Beige	#F5F5DC
Белый	White	#FFFFFF
Белый дымчатый	Whitesmoke	#F5F5F5
Бирюзовый	Turquoise	#40E0D0
Бисквитный	Bisque	#FFE4C4
Бледно-красно-фиолетовый	Palevioletred	#DB7093
Бледно-розово-лиловый	Lavenderblush	#FFF0F5
Бледно-бирюзовый	Paleturquoise	#AFEEEE
Бледно-зеленый	Palegreen	#98FB98
Бледно-золотой	Palegoldenrod	#EEE8AA
Бледно-лиловый	Lavender	#E6E6FA
Блекло-серо-голубой	Cadetblue	#5F9EA0
Блекло-голубой	Aliceblue	#F0F8FF
Болотный	Moccasin	#FFF4B5
Васильковый	Cornflowerblue	#6495ED
Весенний зеленый	Springgreen	#00FF7F
Голубовато-стальной	Steelblue	#4682B4
Голубой	Blue	#0000FF
Грязно-серый	Navajowhite	#FFDEAD
Дыни	Papyawhip	#FFEFD5
Желтовато-коричневый	Tan	#D2B48C
Желто-зеленый	Yellowgreen	#9ACD32
Желтый	Yellow	#FFFF00
Зеленовато-известковый	Limegreen	#32CD32
Зелено-желтый	Greenyellow	#ADFF2F
Зеленый	Green	#008000
Зеленый травы	Lawngreen	#7CFC00
Золотой	Gold	#FFD700
Индиго	Indigo	#4B0082
Коралловый	Coral	#FF7F50
Коричневый	Brown	#A52A2A
Коричневый	Peru	#CD853F
Королевский голубой	Royalblue	#4169E1
Красного золота	Goldenron	#DAA520
Красно-оранжевый	Orangered	#FF4500

Цвет	Имя цвета	Значение
Красный	Red	#FF0000
Лазурь	Azure	#F0FFFF
Лесной зелени	Forestgreen	#228B22
Лимонный	Lemonchiffon	#FFFACD
Льняной	Linen	#FAF0E6
Малиновый	Crimson	#DC143C
Морской волны	Seagreen	#2E8B57
Морской пены	Deashell	#FFF5EE
Мятно-кремовый	Mintcream	#F5FFFA
Небесно-голубой	Skyblue	#87CEEB
Ночной синий	Midnightblue	#191970
Огнеупорного кирпича	Firebrick	#B22222
Оливковый	Olive	#808000
Оранжево-розовый	Maroon	#800000
Оранжево-розовый	Salmon	#FA8072
Оранжевый	Orange	#FFA500
Орхидейный	Orchid	#DA70D6
Охра	Sienna	#A0522D
Персиковый	Peachpuff	#FFDAB9
Пурпурный	Purple	#800080
Пшеничный	Wheat	#F5DEB3
Розово-коричневый	Rosybrown	#BC8F8F
Розовый	Pink	#FFC0CB
Рыже-коричневый	Sandybrown	#F4A460
Свежего меда	Honeydew	#F0FFF0
Светло-голубой	Lightblue	#ADD8E6
Светло-желтый	Lightyellow	#FFFFE0
Светло-зеленый	Lightgreen	#90EE90
Светло-коралловый	Lightcoral	#F08080
Светло-кремовый	Blanchedalmond	#FFEBCD
Светло-розовый	Lightpink	#FFB6C1
Светло-серый	Lightgray	#D3D3D3
Светло-стальной	Lightsteelblue	#B0C4DE
Светло-фиолетовый	Blueviolet	#8A2BE2
Светло-циановый	Lightcyan	#E0FFFF
Светлый красно-золотистый	Lightgoldenrodyellow	#FAFAD2
Светлый морской волны	Lightseagreen	#20B2AA
Светлый небесно-синий	Lightskyblue	#87CEFA
Светлый оранжево-розовый	Lightsalmon	#FFA07A
Светлый серо-фиолетовый	Gainsboro	#DCDCDC
Светлый синевато-серый	Lightslategray	#778899
Серебристый	Silver	#C0C0C0

Цвет	Имя цвета	Значение
Серовато-синий	Slateblue	#6A5ACD
Серый	Gray	#BEBEBE
Синевато-серый	Slategray	#708090
Синий	Aque	#00FFFF
Сливовый	Plum	#DDA0DD
Слоновой кости	Ivory	#FFFFFF0
Снежный	Snow	#FFFAFA
Старого дерева	Burlywood	#DEB887
Старого коньяка	Oldlace	#FDF5E6
Старой кожи	Saddlebrown	#8B4513
Темно-красно-синий	Darkgoldenron	#B8860B
Темно-бирюзовый	Darkturquoise	#00CED1
Темно-голубой	Darkblue	#00008B
Темно-зеленый	Cornsilk	#FFF8DC
Темно-зеленый	Darkgreen	#006400
Темно-красный	Darkred	#8B0000
Темно-оливковый	Darkolivegreen	#556B2F
Темно-оранжево-розовый	Darksalmon	#E9967A
Темно-оранжевый	Darkorange	#FF8C00
Темно-орхидейный	Darkorchid	#9932CC
Темно-розовый	Deeppink	#FF1493
Темно-серый	Darkgrey	#A9A9A9
Темно-синий	Navy	#000080
Темно-фиолетовый	Darkviolet	#9400D3
Темный морской волны	Darkseagreen	#8FBC8F
Темный небесно-синий	Deepskyblue	#00BFFF
Темный серовато-синий	Darkslategrey	#483D8B
Темный синевато-серый	Darkslategray	#2F4F4F
Темный фуксин	Darkmagenta	#8B008B
Темный хаки	Darkkhaki	#BDB76B
Темный циан	Darkcyan	#008B8B
Томатный	Tomato	#FF6347
Туманно-белый	Ghostwhite	#F8F8FF
Туманно-голубой	Powderblue	#B0E0E6
Туманно-розовый	Mistyrose	#FFF4E1
Тускло-васильковый	Dodgerblue	#1E90FF
Тускло-коричневый	Olivedrab	#6B8E23
Тускло-серый	Dimgray	#696969
Умеренно-аквамариновый	Mediumaquamarine	#66CDAA
Умеренно-бирюзовый	Mediumturquoise	#48D1CC
Умеренно-голубой	Mediumblue	#0000CD
Умеренно-орхидейный	Mediumorchid	#BA55D3

Цвет	Имя цвета	Значение
Умеренно-пурпурный	Mediumpurple	#9370DB
Умеренный красно-фиолетовый	Mediumvioletred	#C71585
Умеренный морской волны	Mediumseagreen	#3CB371
Умеренный серовато-синий	Mediumslateblue	#7B68EE
Умеренный синевато-серый	Mediumspringgreen	#00FA9A
Фиолетовый	Violet	#EE82EE
Фисташковый	Chartreuse	#7FFF00
Фуксин	Magenta	#FF00FF
Фуксия	Fuchsia	#FF00FF
Хаки	Khaki	#F0D58C
Цвет извести	Lime	#00FF00
Цветочно-белый	Floralwhite	#FFFAF0
Циан	Cyan	#00FFFF
Чайный	Teal	#008080
Черный	Black	#000000
Чертополоха	Thistle	#D8BFD8
Шоколадный	Chocolate	#D2691E
Ярко-красный	Indianred	#CD5C5C
Ярко-розовый	Hotpink	#FF69B4

Приложение II

Таблица замены специальных символов

В спецификации HTML указано большое количество спецсимволов, но только некоторые из них используются. Наиболее часто используемые спецсимволы приведены в этой таблице.

При замене символов, указанных в первом столбце, можно использовать имя, указанное в третьем столбце, либо код, указанный в четвертом столбце.

Символ	Описание	Имя	Код
Пробелы и тире			
	неразрывный пробел	 sp;	
	узкий пробел (en-шириной в букву N)	 	 
	широкий пробел (em-шириной в букву M)	 	 
-	узкое тире (en-тире)	–	–
—	широкое тире (em -тире)	—	—
	мягкий перенос	­	­
Кавычки			
'	штрих (минуты, футы)	′	′
"	двойной штрих (секунды, дюймы)	″	″
"	прямая кавычка	"	"
“	кавычка-лапка левая	“	“
”	кавычка-лапка правая верхняя	”	”
„	кавычка-лапка правая нижняя	„	„
«	кавычка-елочка открывающая	«	«
»	кавычка-елочка закрывающая	»	»
<	одинарная угловая кавычка открывающая	‹	‹
>	одинарная угловая кавычка закрывающая	›	›
‘	левая верхняя одинарная кавычка	‘	‘
’	правая верхняя одинарная кавычка	’	’
,	правая нижняя одинарная кавычка	‚	‚
Знаки валют			
¢	знак цента	¢	¢
£	знак фунта	£	£
€	знак евро	€	€
¥	знак йены и юаня	¥	¥
¤	знак валюты	¤	¤
f	знак флорина	ƒ	ƒ
Прочие символы			
§	параграф	§	§

Символ	Описание	Имя	Код
°	градус	°	°
...	многоточие	…	…
©	копирайт	©	©
®	знак зарегистрированной торговой марки	®	®
™	знак торговой марки	™	™
μ	знак микро	µ	µ
‰	тысячная доля	‰	‰
&	амперсant	&	&
–	надчеркивание	‾	‾
’	знак ударения	´	´
	вертикальный пунктир	¦	¦
¶	знак абзаца	¶	¶
¡	перевернутый восклицательный знак	¡	¡
¿	перевернутый вопросительный знак	¿	¿
Маркеры			
•	простой маркер	•	•
·	средняя точка	·	·
†	крестик	†	†
‡	двойной крестик	‡	‡
♠	пики	♠	♠
♣	трефы	♣	♣
♥	червы	♥	♥
♦	бубны	♦	♦
◇	ромб	◊	◊
Стрелки			
←	стрелка влево	←	←
↑	стрелка вверх	↑	↑
→	стрелка вправо	→	→
↓	стрелка вниз	↓	↓
↔	стрелка влево и вправо	↔	↔
Знаки арифметических и математических операций			
×	знак умножения	×	×
÷	знак деления	÷	÷
/	дробная черта	⁄	⁄
–	знак минус	−	−
<	знак меньше	<	<
>	знак больше	>	>

Символ	Описание	Имя	Код
\leq	меньше или равно	≤	≤
\geq	больше или равно	≥	≥
\approx	приблизительно равно (асимптотически равно)	≈	≈
\neq	не равно	≠	≠
\equiv	совпадает с	≡	≡
\pm	плюс-минус	±	±
$\frac{1}{4}$	одна четвёртая	¼	¼
$\frac{1}{2}$	одна вторая	½	½
$\frac{3}{4}$	три четверти	¾	¾
1	единица в верхнем индексе	¹	¹
2	два в верхнем индексе (квадрат)	²	²
3	три в верхнем индексе (куб)	³	³
$\sqrt{\quad}$	квадратный корень (радикал)	√	√
∞	знак бесконечность	∞	∞
Σ	знак суммирования	∑	∑
Π	знак произведения	∏	∏
∂	частичный дифференциал	∂	∂
\int	интеграл	∫	∫
f	знак функции	ƒ	ƒ
Буквы латинского алфавита			
æ	латинские строчные буквы æ	æ	æ
ð	латинские строчные символы eth	ð	ð
Θ	заглавная тета	Θ	Θ
Σ	заглавная сигма	Σ	Σ
Ω	заглавная омега	Ω	Ω
α	строчная альфа	α	α
β	строчная бета	β	β
γ	строчная гамма	γ	γ
δ	строчная дельта	δ	δ
ε	строчная эпсилон	ε	ε
ζ	строчная дзета	ζ	ζ
η	строчная эта	η	η
θ	строчная тета	θ	θ
ι	строчная иота	ι	ι
κ	строчная каппа	κ	κ
λ	строчная лямбда	λ	λ
μ	строчная мю	μ	μ

Символ	Описание	Имя	Код
ν	строчная ню	ν	ν
ξ	строчная кси	ξ	ξ
ο	строчная омикрон	ο	ο
π	строчная пи	π	π
ρ	строчная ро	ρ	ρ
ς	строчная сигма(final)	ς	ς
σ	строчная сигма	σ	σ
τ	строчная тау	τ	τ
υ	строчная ипсилон	υ	υ
φ	строчная фи	φ	φ
χ	строчная хи	χ	χ
ψ	строчная пси	ψ	ψ
ω	строчная омега	ω	ω

Приложение III

Параметры форматирования, доступные при использовании CSS

Параметр	Возможные значения	Описание атрибута
Работа с текстом		
letter-spacing	Например: letter-spacing:5px	Расстояние между буквами (межсимвольный интервал). В пикселях (px) или пунктах (pt)
line-height	Например: line-height:3%	Высота текущей строки (междустрочный интервал). В пикселях (px), пунктах (pt) или процентах (%) относительно текущего размера шрифта
text-align	<ul style="list-style-type: none"> • left – по левому краю, • right – по правому краю, • center – по центру, • justify – по ширине. Например: text-align: left	Выравнивание текста
text-indent	Например: text-indent:5pt	Отступ первой строки. В пикселях (px), пунктах (pt) или процентах (%) относительно ширины родительского элемента
vertical-align	<ul style="list-style-type: none"> • baseline — выравнивание по образцу родительского элемента • super — переводит элемент в верхний регистр • sub — нижний регистр • text-top — выравнивает по верху текста, набранного родительским шрифтом • text-bottom — выравнивает по низу текста, набранного родительским шрифтом • top — по самому верху 	Вертикальное положение текущего элемента

Параметр	Возможные значения	Описание атрибута
	<ul style="list-style-type: none"> bottom — по самому низу Например: vertical-align:top	
text-decoration	<ul style="list-style-type: none"> underline – подчеркнутый overline – надчеркнутый line-through – перечеркнутый Например: text-decoration:overline	«Украшение» текста
text-transform	<ul style="list-style-type: none"> Capitalize - каждое слово начинается с большой буквы UPPERCASE - каждая буква текста становится заглавной lowercase - каждая буква текста становится маленькой Например: text-transform:capitalize	Изменение текста
Работа с фоном и цветом		
color	Например: color:#AF3399	Задаёт цвет (любой из трех способов задания)
background-color	Например: background-color:red	Цвет фона элемента
background-image	Например: background-image:URL(fon.jpg)	Задаёт фоновое изображение
background	Например: background: black fixed 50% 0%	Объединяет все вышеперечисленные свойства
background-attachment	<ul style="list-style-type: none"> fixed – не будет прокручиваться, scroll – будет прокручиваться. Например: background-attachment:fixed	Определяет, будет ли прокручиваться фон
background-position	Например: background-position: 10px 5%	Определяет начальное положение фоновой картинки по горизонтали и вертикали. В пикселях (px) или процентах (%)

Параметр	Возможные значения	Описание атрибута
background-repeat	<ul style="list-style-type: none"> • repeat-x – повтор по горизонтали, • repeat-y – повтор по вертикали, • repeat (по умолчанию) – по всем направлениям, • no-repeat – не повторяется. Например: background-repeat:repeat-x	Определяет, повторяется ли фоновая картинка
Работа со шрифтом		
font-family	Например: font-family:Arial	Задаёт название шрифта
font-style	<ul style="list-style-type: none"> • normal – без изменений • italic – курсив Например: font-style:italic	Задаёт стиль начертания
font-variant	<ul style="list-style-type: none"> • normal - без изменений • small-caps - заменяет все буквы на большие. Например: font-variant:small-caps	Задаёт прописные (заглавные буквы)
font-weight	<ul style="list-style-type: none"> • normal - без изменений • bold – жирный • любое значение от 100 до 900 Например: font-weight:bold	Задаёт жирность текста
font-size	<ul style="list-style-type: none"> • smaller • larger • числовое значение Например: font-size:30px	Задаёт размер шрифта. В пикселях (px), пунктах (pt) или процентах (%)
font	Например: font: italic bold Arial 12pt	Объединяет все описанные выше свойства
Работа с рамкой		
border-top-width, border-right-width, border-bottom-width, border-left-width	Например: border-top-width:100px	Толщина верхней, правой, нижней или левой рамки соответственно. В пикселях (px) или

Параметр	Возможные значения	Описание атрибута
		процентах (%)
border-color	Например: border-color:green	Цвет рамки
border-style	<ul style="list-style-type: none"> • none - рамка отсутствует • dotted - рамка представляется мелким пунктиром. • dashed - рамка представляется крупным пунктиром • solid – сплошная рамка • double – рамка в виде двух сплошных линий. • groove - рамка выглядит вдавленной • ridge - рамка выглядит выпуклой. Например: border-style:double	Стиль рамки
Работа с отступом		
margin-top, margin-right, margin-bottom, margin-left.	Например: margin-top:100px	Отступ сверху, справа, снизу и слева соответственно у блочных элементов (таблицы, слоя, картинки, тэга <BODY>). В пикселях (px) или процентах (%)
margin	Например: margin:10%	Определяет отступ во всех направлениях
padding-top, padding-right, padding-bottom, padding-left	Например: padding-bottom:100px	Отступ сверху, справа, снизу и слева соответственно у любого элемента (таблицы, слоя, параграфа и др.). В пикселях (px) или процентах (%)
padding	Например: padding:10px	Отступ во всех направлениях

Параметр	Возможные значения	Описание атрибута
Работа с размером элемента		
width	Например: width:10%	Ширина элемента. В пикселях (px) или процентах (%)
height	Например: height:100px	Высота элемента. В пикселях (px) или процентах (%)
Позиционирование элемента (слоя)		
position	<ul style="list-style-type: none"> absolute — координаты будут вычисляться от верхнего левого угла документа relative — координаты будут вычисляться от верхнего левого угла родительского элемента static — элемент не будет прокручиваться при прокручивании страницы Например: position:absolute	Устанавливает, каким образом определяется положение элемента на странице
top	Например: top:10px	Устанавливает вертикальную координату элемента относительно верхнего края. В пикселях (px) или в процентах (%) от высоты родительского элемента
bottom	Например: bottom:10%	Устанавливает вертикальную координату элемента относительно нижнего края. В пикселях (px) или в процентах (%) от высоты родительского элемента
left	Например: left:10%	Устанавливает горизонтальную координату элемента относительно левого

Параметр	Возможные значения	Описание атрибута
		края. В пикселях (px) или в процентах (%) от ширины родительского элемента
right	Например: right:10px	Устанавливает горизонтальную координату элемента относительно правого края. В пикселях (px) или в процентах (%) от ширины родительского элемента
Видимость элемента (слоя)		
visibility	<ul style="list-style-type: none"> • visible — элемент будет виден • hidden — элемент не будет виден Например: visibility:visible	Устанавливает, будет ли виден элемент
z-index	Например: z-index:-3	Указывает, в каком порядке элементы будут перекрываться. Элементы с более высоким z-index будут перекрывать элементы с более низким. Возможны любые значения, включая отрицательные. В случае отрицательных значений элементы будут закрыты обычным текстом; с положительным значением — будут лежать «перед» обычным текстом

Приложение IV

Список тэгов и параметров, рассмотренных в пособии

Тэг	Параметры	Назначение
Структура HTML-документа		
<HTML> и </HTML>		границы HTML-документа
<HEAD> и </HEAD>		голова HTML-документа
<BODY> и </BODY>		тело HTML-документа
	bbgcolor	цвет фона страницы
	text	цвет текста всей страницы
	link	цвет ссылки, еще не посещенной
	vlink	цвет ссылки, уже посещенной
	alink	цвет активной ссылки
	background	фоновый рисунок страницы
	leftmargin, rightmargin, topmargin, bottommargin	задают левое, правое, верхнее и нижнее поля страницы соответственно (в % или px)
<TITLE> и <TITLE>		заголовок (название) странички
Форматирование текста		
<P> и </P>		абзац
	title	краткое описание параграфа
	align	выравнивание абзаца (возможные значения <i>center, left, right, justify</i>)
<Hx> и </Hx>, где x=1..6		встроенный заголовок
 		перенос строки
<BLOCKQUOTE> и </BLOCKQUOTE>		отступ текста
<PRE> и </PRE>		предварительно форматированный текст
<CITE> и </CITE>		выделение цитаты
 и 		выделение важных фрагментов полужирным шрифтом

Тэг	Параметры	Назначение
<VAR> и </VAR>		для отметки имен переменных
<STRIKE> и </STRIKE>		зачеркивание текста
 и 		полужирный шрифт
<I> и </I>		курсив
<U> и </U>		подчеркнутый
<S> и </S>		перечеркивание
<TT> и </TT>		шрифт фиксированной ширины
_и		нижний индекс
^и		верхний индекс
		работа со шрифтом
	size	размер шрифта
	face	тип шрифта
	color	цвет
<MARQUEE> и </MARQUEE>		бегущая строка
	bgcolor	цвет фона
	height	высота фоновой полосы (в px или %)
	width	ширина полосы бегущей строки (в px или %)
	direction	направление бегущей строки (возможные значения <i>left</i> , <i>right</i> , <i>up</i> , <i>down</i>)
	behavior	поведение бегущей строки (возможные значения <i>scroll</i> , <i>slide</i> , <i>alternate</i>)
	hspace, vspace	горизонтальный и вертикальный отступы у бегущей строки (в px или %)
	loop	количество переходов строки по экрану
	scrollamount	скорость движения строки (в px в секунду)
 и 		нумерованный список
	start	число, с которого начать нумерацию
	type	тип нумерации

Тэг	Параметры	Назначение
		(возможные значения i, I, a, A,)
 и 		маркированный список
	type	вид маркера (возможные значения <i>circle</i> , <i>square</i> , <i>disc</i>)
 и 		элемент списка
Графика в HTML		
		вставка рисунка
	src	путь к рисунку
	width	ширина рисунка (в px)
	height	высота рисунка (в px)
	align	положение рисунка относительно текста (возможные значения <i>top</i> , <i>middle</i> , <i>center</i> , <i>bottom</i> , <i>left</i> , <i>right</i>)
	hspace, vspace	горизонтальный и вертикальный отступы текста от изображения (в px)
	border	граница картинки (по умолчанию 1)
	alt	альтернативный текст
<HR>		горизонтальная полоса
	width	длина линии (в % или px)
	size	ширина линии (в px)
	align	положение линии на странице (возможные значения <i>right</i> , <i>left</i> , <i>center</i>)
	color	цвет линии
Работа с таблицами		
<TABLE> и </TABLE>		задание таблицы
	border	ширина границы таблицы (в px)
	bordercolor	цвет границы таблицы
	width	ширина всей таблицы (в px или %)
	height	высота всей таблицы (в px или %)

Тэг	Параметры	Назначение
	bgcolor	цвет фона под таблицей
	background	фоновый рисунок таблицы
	align	положение таблицы на странице (возможные значения <i>left</i> , <i>center</i> , <i>right</i>)
	cellpadding	отступ текста от границ ячеек
	cellspacing	отступ табличных объектов (ячейки, строки) друг от друга
<CAPTION> и </CAPTION>		название таблицы
	align	положение названия таблицы (возможные значения <i>bottom</i> , <i>left</i> , <i>right</i> , <i>top</i>)
<TR> и </TR>		строка таблицы
	align	выравнивание внутри строки (возможные значения: <i>left</i> , <i>right</i> , <i>center</i>);
	valign	вертикальное выравнивание внутри строки (возможные значения: <i>top</i> , <i>bottom</i> , <i>middle</i>);
	bgcolor	цвет фона внутри строки
	background	фоновый рисунок строки
<TH> и </TH>		ячейка-заголовок строки или столбца
<TD> и </TD>		ячейка данных
	colspan, rowspan	растягивание ячейки на несколько столбцов или несколько строк
	height	высота ячейки (в px или %)
	width	ширина ячейки (в px или %)

Тэг	Параметры	Назначение
	nowrap	запрет или разрешение на перенос строк внутри ячейки
Ссылки		
текст ссылки		задание ссылки
текст метки		задание метки
текст ссылки.		задание ссылки на метку
	link	цвет ссылки, еще не посещенной
	alink	цвет активной ссылки
	vlink	цвет ссылки, уже посещенной
	title	текст подсказки, которая всплывает при наведении на ссылку
	target	способ открытия документа, на который ведет ссылка (возможные значения <i>_self</i> , <i>_parent</i> , <i>_top</i>)
Фреймы		
<FRAMESET> и </FRAMESET>		задание фреймовой структуры
	rows	количество горизонтальных фреймов
	cols	количество вертикальных фреймов
	frameborder	наличие рамки, обрамляющей фреймы (возможные значения "yes" или "no")
	border	толщина рамки, обрамляющей фреймы (в пх или %)

Тэг	Параметры	Назначение
	bordercolor	цвет рамки, обрамляющей фреймы
<FRAME>...		задание фрейма
	src	путь к файлу, который отображается в данном фрейме
	name	имя фрейма
	scrolling	наличие полосы прокрутки во фрейме (значения «yes» или «no»)
	noresize	разрешение или запрет на изменение размера фрейма
	marginwidth, marginheight	отступ содержимого внутри фрейма по горизонтали и вертикали соответственно (в px)
<IFRAME>	Параметры, свойственные тэгу <FRAME> и параметры, свойственные тэгу 	задание плавающего фрейма
Работа с формами		
<FORM> и </FORM>		задание формы
<INPUT>		для ввода текстовой информации, файлов или управляющей информации
	name	имя поля (имя переменной, которая будет отправляться на сервер)
	size	визуальный размер поля ввода на экране в символах (по умолчанию size=20)
	maxlength	количество символов, которое пользователи могут ввести в поле ввода

Тэг	Параметры	Назначение
	value	определяет начальное (по умолчанию) значение поля либо значение, которое нужно отправить на сервер
	type	определяет тип поля ввода (возможные значения – <i>text, password, checkbox, radio, reset, submit, image, file</i>)
<TEXTAREA> и </TEXTAREA>		текстовое поле в несколько строк
	name	наименование поля
	cols	число колонок (символов) в текстовой области
	rows	число видимых строк в текстовой области
<SELECT> и </SELECT>		список значений
	name	имя объекта SELECT
	multiple	позволяет выбрать более чем одно наименование
	single	из списка можно выбрать только один вариант
	size	определяет число пунктов, видимых для пользователя
<OPTION>		элементы списка
	selected	элемент выбран по умолчанию
	value	значение, которое передается на сервер для обработки. По умолчанию значение поля равно элементу <OPTION>
Каскадные таблицы стилей (подробно о CSS смотрите в Приложении III)		
<STYLE> и </STYLE>		описание стилей внутри HTML-документа
<ТЭГ style="описание		применение стиля к

Тэг	Параметры	Назначение
стиля”>		определенному тэгу
<ТЭГ class=”название класса”>		применение класса к определенному тэгу
<DIV> и </DIV>		задание слоя

Библиографический список

1. CSS - Cascading Style Sheets [Электронный ресурс]. - Internet:
<http://londomalary.chat.ru/css2.html>
2. HTML [Электронный ресурс]. - Internet:
<http://web-script.narod.ru/articles/HTML/HTML.html>
3. HTML для начинающих [Электронный ресурс]. - Internet:
<http://www.wsa.ru/enc/web/html/ok/lesson0.htm>
4. Браун, М. HTML в подлиннике: электронный учебник [Электронный ресурс] / М. Браун, Д. Ханикатт.
5. Спецификация HTML 4.01 (W3C Recommendation 24 December 1999) HTML 4.01 Specification [Электронный ресурс]. – Internet:
<http://www.w3.org/TR/1999/REC-html401-19991224>
6. Учебник HTML [Электронный ресурс]. - Internet:
<http://excellentweb.amillo.net/index.php?page=html/index.php>
7. Учебник HTML [Электронный ресурс]. - Internet:
<http://srez.ru/index.php?n=teach/html/index>

Оглавление

Введение	3
Глава 1. Структура HTML-документа.....	5
Глава 2. Форматирование HTML-документа	10
Глава 3. Графика в HTML-документе	28
Глава 4. Работа с таблицами.....	35
Глава 5. Гиперссылки.....	43
Глава 6. Фреймы	52
Глава 7. Создание форм	61
Глава 8. Каскадные таблицы стилей	70
Проектное задание. Разработка сайта «Моя личная страничка»	82
Приложения	84
Приложение I. Web-палитра цветов	84
Приложение II. Таблица замены специальных символов	88
Приложение III. Параметры форматирования, доступные при использовании CSS	92
Приложение IV. Список тэгов и параметров, рассмотренных в пособии.....	98
Библиографический список.....	106
Оглавление	107

Учебное издание

Васенина Екатерина Александровна

Основы языка HTML

Редактор О. Коробкова

Подписано в печать 24.06.2005 г.

Формат 60×84/16

Бумага тип.

Усл. печ. л. 6,7

Тираж 100 экз.

Издательство Вятского государственного гуманитарного университета,
610002, г. Киров, ул. Красноармейская, 26