

А.Н. Маслобоев

**Базовые возможности
языка HTML
по разработке и оформлению
Web-страниц**

Учебно-методическое пособие



**Санкт-Петербург
2016**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ И ДИЗАЙНА»**

ВЫСШАЯ ШКОЛА ТЕХНОЛОГИИ И ЭНЕРГЕТИКИ

А.Н. Маслобоев

**Базовые возможности
языка HTML
по разработке и оформлению
Web-страниц**

Учебно-методическое пособие

**Санкт-Петербург
2016**

УДК 681.3 (075)

М 316

ББК 22.18я7

Маслобоев А.Н. Базовые возможности языка HTML по разработке и оформлению Web-страниц: учебно-методическое пособие/ ВШТЭ СПбГУПТД. — СПб., 2016. — 87 с.

В настоящем учебно-методическом пособии рассматриваются вопросы использования возможностей языка HyperText Markup Language (HTML) для разработки и оформления Web-страниц. Пособие содержит необходимые теоретические сведения, примеры HTML-кода Web-страниц с подробными комментариями. К каждой главе приведены вопросы для самоконтроля.

Пособие предназначено для бакалавров направления подготовки 01.03.02 «Прикладная математика и информатика», обучающихся по дисциплинам «Web-страницы» и «Современные издательские системы».

Рецензенты:

профессор кафедры ИИТиСУ ВШТЭ СПбГУПТД, д-р техн. наук
Г.А. Кондрашкова

профессор кафедры ИИСТ СПбГЭТУ «ЛЭТИ», д-р техн. наук
Е.М. Антонюк

Рекомендовано к изданию Редакционно-издательским советом ВШТЭ СПбГУПТД в качестве учебно-методического пособия.

© Маслобоев А.Н., 2016

© Высшая школа технологии и
энергетики СПбГУПТД, 2016

Оглавление

Введение	4
ГЛАВА 1. Основные операции с текстом.....	5
1.1. Содержание стандартной Web-страницы.....	6
1.2. Структурирование Web-страницы	9
1.3. Форматирование шрифта	11
1.4. Форматирование абзацев. Использование литералов.....	17
Вопросы для самоконтроля.....	21
ГЛАВА 2. Создание списков	22
2.1. Нумерованные списки	22
2.2. Маркированные списки.....	26
2.3. Списки определений.....	30
2.4. Многоуровневые списки	32
Вопросы для самоконтроля.....	36
ГЛАВА 3. Использование графических элементов.....	37
3.1. Основные форматы графических файлов	37
3.2. Вставка иллюстраций в текст страницы.....	38
3.3. Использование графики в качестве фона страницы	45
Вопросы для самоконтроля.....	48
ГЛАВА 4. Использование гиперссылок	49
4.1. Текстовые гиперссылки	49
4.2. Графические гиперссылки	58
4.3. Закладки и внутристраничные гиперссылки	59
Вопросы для самоконтроля.....	67
ГЛАВА 5. Работа с таблицами.....	68
5.1. Создание простой таблицы	68
5.2. Вложенные таблицы	79
Вопросы для самоконтроля.....	87
Библиографический список	87

Введение

В XXI веке существование и дальнейшее развитие человеческой цивилизации немыслимо без широкого использования информационных технологий. В настоящее время не осталось ни одной сферы человеческой деятельности, в которой бы не нашли свое применение разнообразные компьютеры в диапазоне от суперкомпьютеров, решающих глобальные задачи, до различных миниатюрных мобильных устройств, ставших неизменными спутниками современного человека, где бы он ни находился.

Важнейшей составной частью современных информационных технологий являются сетевые технологии. Благодаря наличию глобальной информационной сети человек, находящийся в любой точке земного шара, может получить доступ к огромному массиву знаний, так как в Интернете хранится практически вся информация, накопленная человечеством за тысячелетия своего развития. Объем циркулирующей в глобальной сети информации с каждым годом и с каждым днем непрерывно увеличивается.

Эта информация хранится на серверах (т.е. компьютерах, постоянно подключенных к Интернету) в виде специальных документов, которые называются Web-страницами. Это название происходит от английского слова **web**, что в переводе означает «сеть». Как правило, Web-страницы объединяются по тематическому принципу в группы, которые называются Web-сайтами (Web-узлами),

На начальном этапе развития Интернета Web-страницы представляли собой текстовые документы, содержащие наряду с обычным текстом гипертекст, т.е. фрагменты текста, оформленные в виде ссылок, позволяющих выполнить переход на другие страницы в сети. Большинство современных Web-страниц, кроме текста, содержат иллюстрации, которые тоже можно использовать как гиперссылки, мультимедийные объекты (аудио- и видеоклипы), а также управляющие элементы, позволяющие сделать страницу интерактивной, взаимодействующей с пользователем.

Для просмотра Web-страниц используются специальные программы-браузеры, а для создания таких страниц применяется специальный язык HTML. Название HTML расшифровывается как HyperText Markup Language, что в переводе с английского означает «язык разметки гипертекста». Этот язык позволяет задать внутреннюю структуру документа, определить способы оформления его контента (содержимого), вставить в текст различные внешние объекты. Знание команд языка HTML и умение свободно им пользоваться является абсолютной необходимостью для специалиста по информационным технологиям.

Задачей настоящего учебного пособия является ознакомление будущих IT-специалистов с основными тегами (командами) языка HTML и демонстрация на конкретных примерах возможностей HTML по разработке Web-страниц, а также по их нестандартному оформлению.

ГЛАВА 1. Основные операции с текстом

Создание новой Web-страницы начинается с создания обычного текстового файла в любом текстовом редакторе. В качестве такого редактора может использоваться, например, программа **Блокнот**, являющаяся стандартным приложением операционной системы **Windows**. В текстовом файле записывается текст Web-страницы на языке **HTML**. Затем этот файл нужно сохранить, указав в диалоговом окне для него расширение **htm** или **html**, но оставив при этом предлагаемый по умолчанию текстовый формат файла, так как это показано на приводимой ниже иллюстрации (рис.1.1).

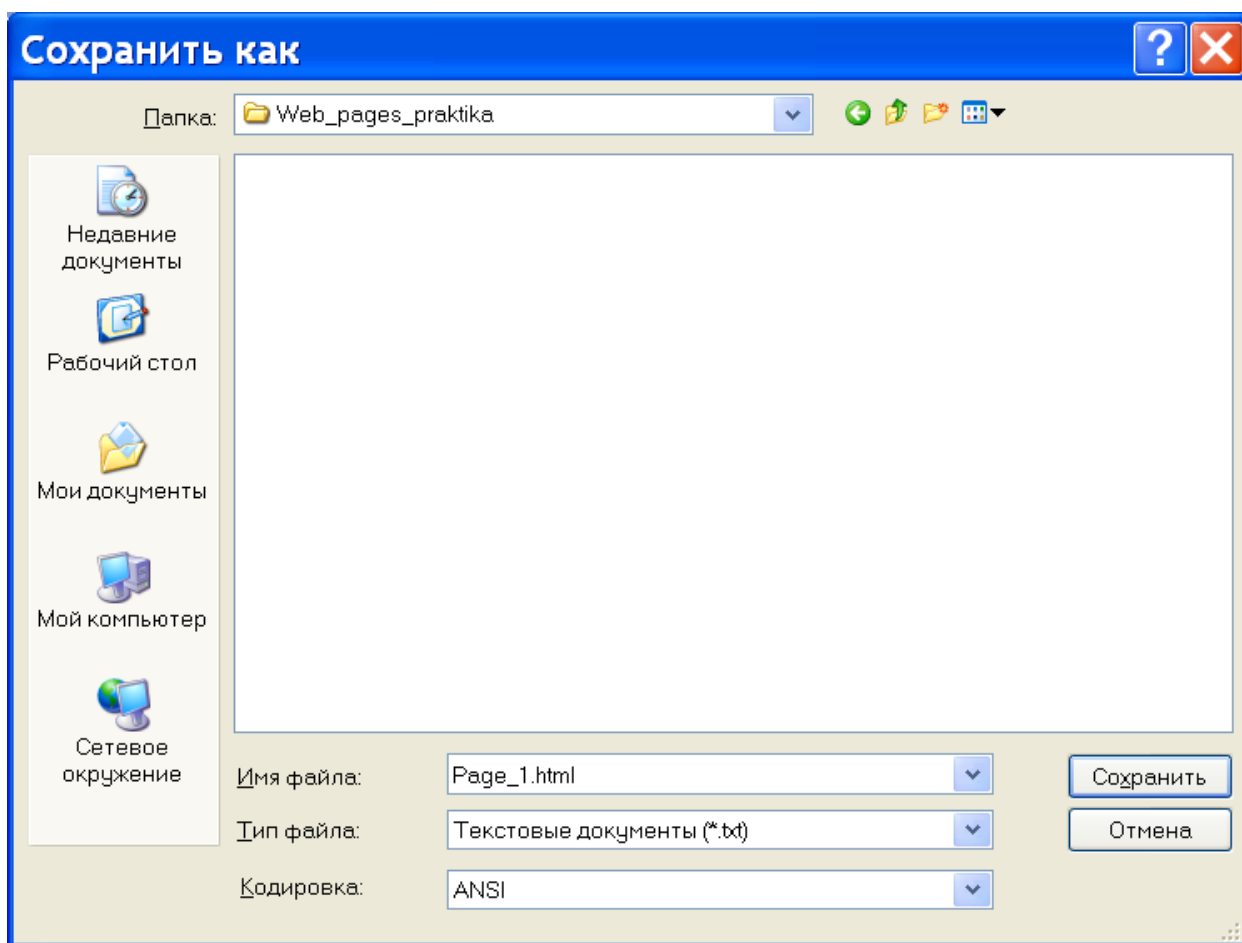


Рис. 1.1. Диалоговое окно, позволяющее сохранить текстовый документ как Web-страницу

Теперь, когда пользователь откроет данный файл на своем компьютере, то он откроется в том браузере, который на этом компьютере по умолчанию является основным. В зависимости от настроек компьютера это может быть **Internet Explorer**, **Mozilla Firefox**, **Opera**, **Google Chrome** или какой-либо другой браузер. Но при этом пользователь увидит только пустую страницу, поскольку структура страницы и какое-либо содержимое пока что отсутствует. Далее рассмотрим, как наполнить такую страницу содержанием.

1.1. Содержание стандартной Web-страницы

Содержание Web-страницы состоит из собственно текста страницы и команд, используемых для его оформления. Данные команды, заключаемые в угловые скобки, называются тегами. Теги подразделяются на парные и одиночные. Парные теги состоят из открывающего и закрывающего тега. Парные теги влияют на фрагмент страницы, расположенный между ними. Для того чтобы отличить открывающий и закрывающий теги друг от друга, в начале закрывающего тега ставят косую черту. Часто для обозначения этого символа употребляют заимствованное из английского языка название «слэш». Пример парного тега:

** Эту фразу необходимо выделить **

Данный тег выделяет заключенную внутри него фразу полужирным шрифтом.

Одиночный тег не влияет непосредственно на находящийся рядом с ним текст, но он также производит определенные действия, направленные на оформление Web-страницы. Пример одиночного тега:

<HR SIZE="3" COLOR="BLUE">

Данный тег создает на Web-странице разделительную горизонтальную линию синего цвета толщиной в 3 пикселя.

Большинство Web-страниц имеют стандартную структуру, создаваемую с помощью специальных тегов. Структура такой стандартной Web-страницы в общем виде выглядит следующим образом:

```
<HTML>
<HEAD>
<TITLE>
.....
</TITLE>
</HEAD>
<BODY>
.....
</BODY>
</HTML>
```

Расшифруем эти теги. Парный тег **<HTML> ...</HTML>** показывает, что данный документ является не просто текстовым файлом, а Web-страницей, созданной на языке **HTML**. Внутри Web-страницы существуют два основных раздела: раздел заголовка и тело документа.

Раздел заголовка содержит информацию, которая описывает документ в целом. Этот раздел создается парным тегом **<HEAD>...</HEAD>**. Одним из тегов, который может находиться внутри данного тега, является тег **<TITLE> ... </TITLE>**, который определяет содержание, отображаемое в

строке заголовка Web-страницы. Основное содержание документа находится в теле документа, которое создается с помощью тега **<BODY>...</BODY>**.

Используем эти теги для создания простой Web-страницы. Вначале необходимо любым способом создать новый текстовый документ. Откроем этот текстовый файл и наберем в нем следующий текст:

```
<HTML>
<HEAD>
<TITLE> Простая Web-страница</TITLE>
</HEAD>
<BODY>
```

Этот документ представляет собой пример простой Web-страницы.

```
<BR>
```

Для его создания в начале был создан обычный текстовый документ, который был затем сохранен в текстовом формате, но с расширением HTML.

```
<BR>
```

Этот документ содержит только заголовок и несколько строк неформатированного текста.

```
<BR>
```

Для просмотра этой страницы можно использовать любой интернет-браузер.

```
</BODY>
```

```
</HTML>
```

Кроме структурных тегов, в этот текстовом документе используется еще один тег **
**. Данный тег является одиночным. Его назначение заключается в том, чтобы завершить текущую строку. Следующий за этим тегом текст начинается с новой строки.

Сохраним текст документа так, как это было показано на рис.1.1. В результате, помимо исходного текстового документа, создается новый документ с расширением **html**. Если на названии этого документа дважды щелкнуть мышью, то он откроется с помощью браузера, который является основным на данном компьютере. Внешний вид созданного документа при просмотре его в браузере **Internet Explorer** приведен на рис.1.2. Обратите внимание, что при просмотре в браузере вы видите только собственно текст, а оформляющие его теги не видны.

Пользователь может просматривать содержимое Web-документа и в других браузерах, если, конечно, они установлены у него на компьютере. Для этого необходимо открыть контекстное меню документа и в этом меню выбрать пункт «Открыть с помощью». В открывшемся вложенном меню следует выбрать название того браузера, в котором необходимо открыть документ.

Например, если на компьютере в качестве основного браузера используется **Internet Explorer**, но установлен также браузер **Mozilla Firefox**, то описанным выше способом можно просмотреть документ и в **FireFox**. Этой возможностью часто пользуются разработчики Web-страниц, чтобы добиться приемлемого вида Web-страницы в различных браузерах. На рис.1.3 показан внешний вид той же простой Web-страницы при просмотре ее в браузере **Mozilla Firefox**.

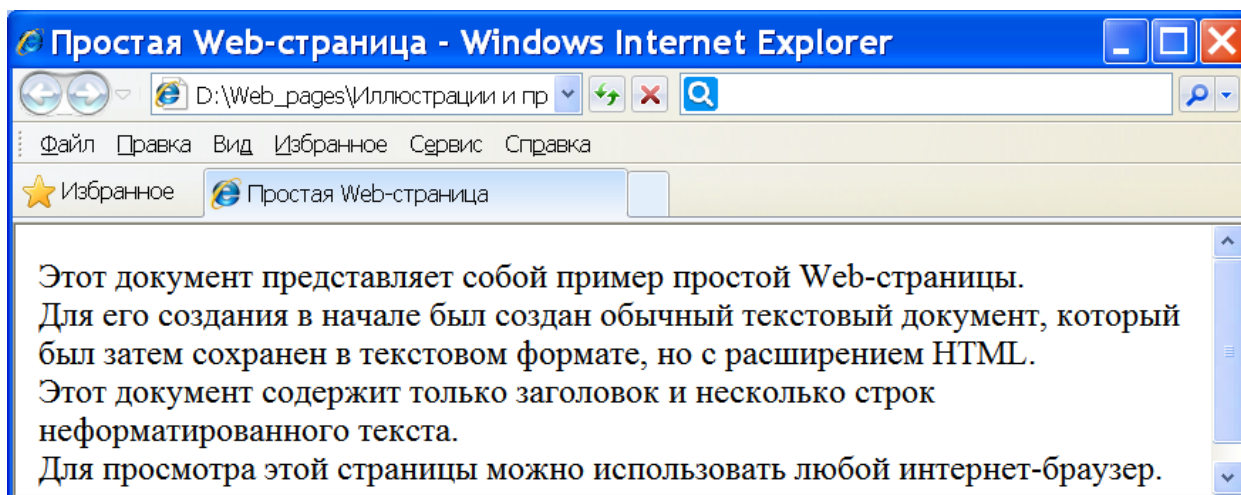


Рис. 1.2. Внешний вид Web-страницы при просмотре ее в браузере Internet Explorer

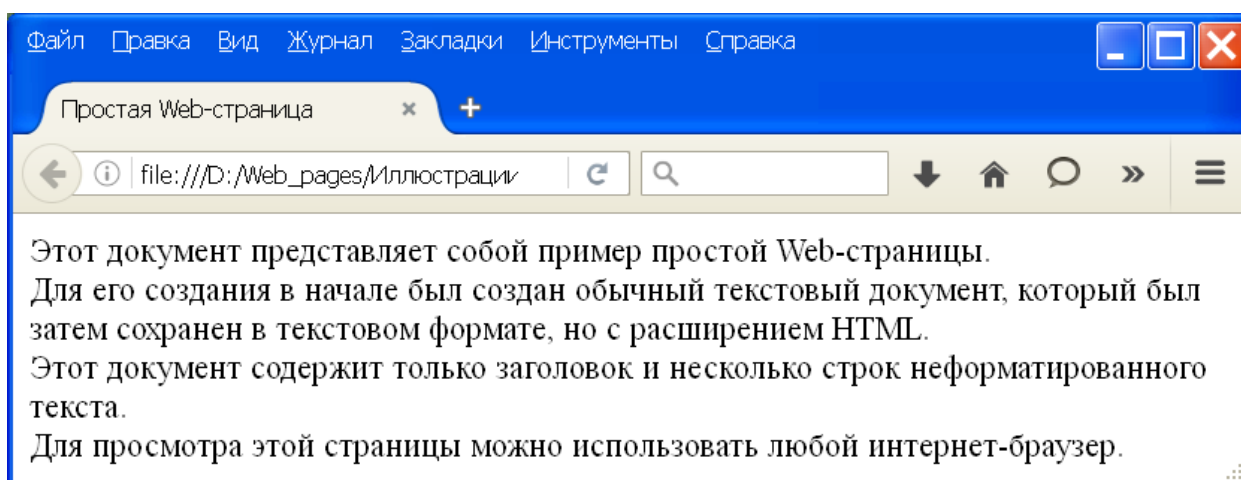


Рис. 1.3. Внешний вид простой Web-страницы при просмотре ее в браузере Mozilla Firefox

Как видно из приведенных выше иллюстраций, представление простой страницы в обоих браузерах практически ничем не отличается.

Естественно, что при написании текста Web-страницы могут появиться ошибки, которые необходимо исправить. Сделать это в браузере невозможно. Для корректировки Web-страницы ее снова нужно открыть с помощью текстового редактора. Для этого следует открыть контекстное меню файла и в нем выбрать пункт «**Открыть с помощью**». Справа от названия пункта

находится треугольная стрелка, указывающая на то, что данный пункт представляет собой вложенное меню. В этом меню в качестве открывающей программы следует выбрать программу **Блокнот** или другой текстовый редактор (если он установлен на компьютере). Тогда в редакторе откроется HTML-код данной страницы, который можно исправлять, улучшать, корректировать нужным образом. После внесения всех необходимых исправлений нужно снова открыть Web-страницу в браузере, чтобы убедиться, что теперь она выглядит должным образом.

Теги языка HTML позволяют не только выводить простой неформатированный текст, как в приведенном выше примере, но и применять различные возможности для его структурирования и оформления. Перечислим основные из этих возможностей:

- а) структурирование текста с помощью заголовков различных уровней;
- б) использование горизонтальных разделительных линий для отделения друг от друга различных разделов страницы;
- в) оформление документа на уровне шрифта, включая изменение гарнитуры шрифта, его начертания, размера, цвета;
- г) изменение фоновой окраски страницы;
- д) форматирование документа на уровне абзаца, включая использование разных способов выравнивания и создание красной строки.

О том, как реализуются все эти способы оформления Web-страниц средствами языка HTML, будет изложено далее в этой главе.

1.2. Структурирование Web-страницы

Для того чтобы обеспечить более четкую внутреннюю структуру Web-страницы, ее разбивают на отдельные смысловые блоки. В начале таких блоков ставят заголовки различных уровней. Всего язык HTML позволяет создавать заголовки 6 уровней. Заголовки создаются с помощью парного тега **<Hn>...</Hn>**, где **n** – номер уровня заголовка.

Наиболее крупные заголовки (заголовки первого уровня) создаются с помощью тега **<H1> ... </H1>**. Менее крупные заголовки (второго уровня) создаются с помощью тега **<H2>...</H2>**, т.е. чем больше номер заголовка, тем меньше его размер и ниже его уровень. Самыми маленькими являются заголовки шестого уровня, создаваемые тегами **<H6>...</H6>**.

Разделительная линия создается с помощью одиночного тега **<HR>**. Данный тег имеет ряд дополнительных параметров, к которым относятся:

- SIZE** – определяет толщину линии, выраженную в пикселях;
- COLOR** – определяет цвет линии, который может задаваться либо в виде текстовой константы, либо в числовой форме;
- WIDTH** – ширина линии, выраженная в процентах от общей ширины страницы.

Дополнительные параметры, называемые также атрибутами, указываются внутри угловых скобок справа от названия самого тега.

Большинство атрибутов имеют определенные значения. Для того чтобы задать значение атрибута, нужно после названия параметра поставить знак равенства, а затем в двойных кавычках указать требуемое значение.

Например, если нужно задать ширину горизонтальной разделительной линии в 5 пикселей, указать синий цвет данной линии и ширину ее в 70 % от общей ширины страницы, то сделать это можно с помощью следующего тега с соответствующим набором дополнительных атрибутов:

```
<HR SIZE="5" COLOR="BLUE" WIDTH="70%">
```

Ниже приводится HTML-код страницы, в которой используются указанные элементы оформления:

```
<HTML>
<HEAD>
<TITLE>
Основы информатики
</TITLE>
</HEAD>
<BODY>
<H1> Базовые понятия информатики </H1>
<HR SIZE="5" COLOR="BLUE">
<H2> Основные определения </H2>
```

Информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии. Важнейшие свойства информации: полнота, достоверность, ценность, актуальность и ясность.

```
<BR>
```

Информатика – это наука, изучающая структуру и наиболее общие свойства информации, ее поиск, хранение, передачу и обработку с применением персональных компьютеров (ПК)

```
<BR>
```

```
<HR SIZE="5" COLOR="BLUE">
```

```
<H2> Основные единицы измерения информации </H2>
```

1 бит – основная единица представления информации

```
<BR>
```

1 байт – основная единица обработки информации

```
<BR>
```

1 байт = 8 бит

```
<BR>
```

1 Килобайт = 1024 байт

```
<BR>
```

1 Мегабайт = 1024 Килобайт

```
<BR>
```

1 Гигабайт = 1024 Мегабайт

```
<BR>  
1 Терабайт = 1024 Гигабайт  
<BR>  
</BODY>  
</HTML>
```

Эта страница содержит 1 заголовок первого уровня и 2 заголовка второго уровня. Каждый из заголовков 2-го уровня открывает определенный текстовый блок. Текстовые блоки отделяются друг от друга горизонтальными разделительными линиями. Для переноса текста на следующую строку используется уже знакомый нам одиночный тег **
**. На рис. 1.4. показан внешний вид страницы при просмотре ее в браузере:

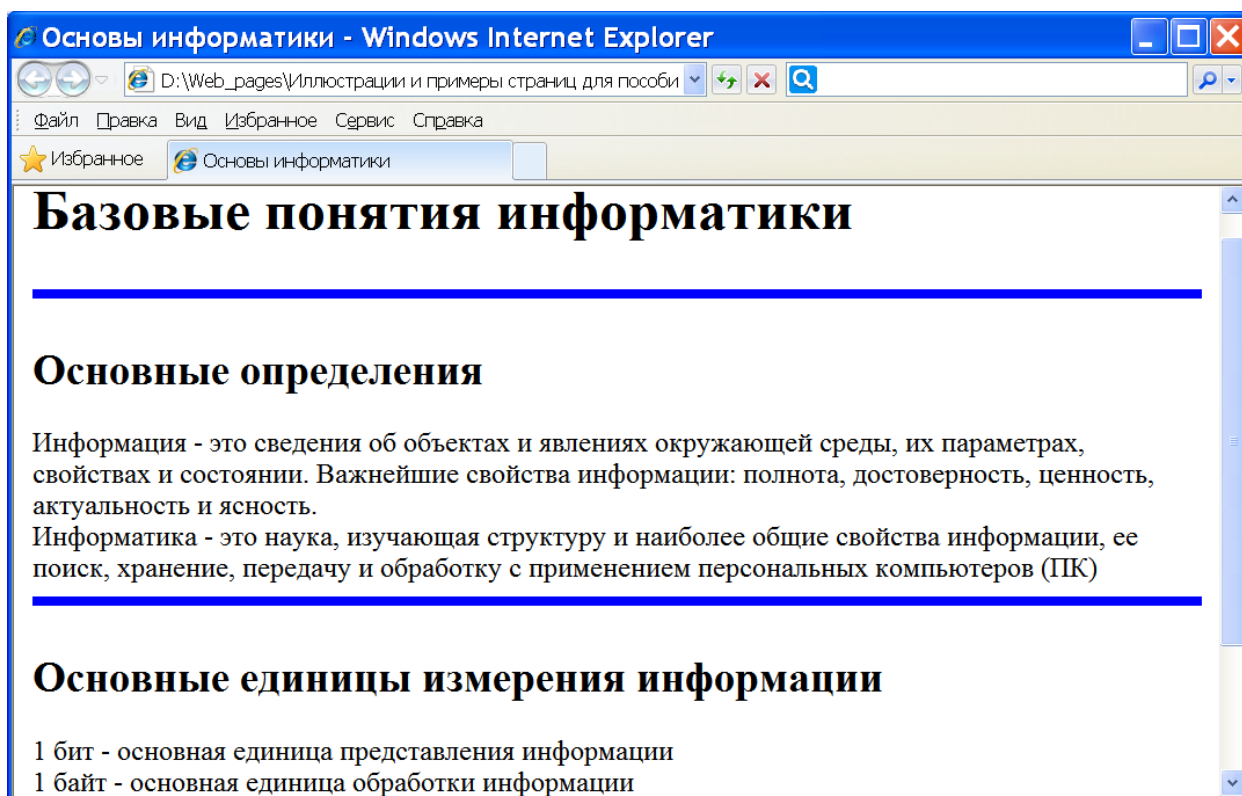


Рис. 1.4. Внешний вид страницы с использованием заголовков разного уровня и горизонтальных разделительных линий при просмотре ее в браузере Internet Explorer

1.3. Форматирование шрифта

В рассмотренном выше примере весь текст, содержащийся на Web-странице, набран одинаковым шрифтом. Но язык HTML предоставляет разработчику широкий набор возможностей по оформлению текста путем изменения параметров шрифта. Эти возможности мы рассмотрим далее.

Основным тегом, используемым для оформления шрифта Web-документа является парный тег ** ... **. Этот тег, как и другие парные теги, влияет на оформление того текстового фрагмента, который

расположен внутри тега, между открывающей и закрывающей его частью. Для оформления шрифта используется ряд дополнительных параметров:

SIZE – определяет относительный размер шрифта;

FACE – задает гарнитуру шрифта;

COLOR – определяет цвет шрифта.

Все перечисленные параметры могут использоваться как по отдельности, так и совместно в одном и том же теге. Рассмотрим особенности каждого из указанных параметров.

Размер шрифта задается параметром **SIZE** не в каких-либо абсолютных единицах, а относительно среднего размера, применяемого в браузере. Этот размер задается в целых единицах от 1 до 7. Средний размер шрифта, используемый по умолчанию, равен 3. Чем меньше численное значение параметра **SIZE**, тем меньше размер шрифта. Таким образом, минимальный размер шрифта равен 1, а максимальный – 7. Например, если на странице содержится тег

```
<FONT SIZE="4"> ...</FONT>
```

то фрагмент шрифта, заключенный внутри тега, будет напечатан шрифтом, который на одну градацию крупнее, чем средний шрифт документа.

Можно указывать размер шрифта и по-другому, относительно среднего размера. Тогда в теге **** значение атрибута **SIZE** указывается со знаком «плюс» или «минус». Если перед числом, заключенным в двойные кавычки стоит «плюс», то шрифт, заключенный внутри тега ****, будет крупнее среднего, если перед числом стоит «минус», то меньше. Например, если в HTML-коде мы видим следующий тег:

```
<FONT SIZE="+2"> ...</FONT>
```

то шрифт, заключенный в этот тег, будет на две градации больше, чем средний. А если тег выглядит таким образом:

```
<FONT SIZE="-1"> ...</FONT>
```

это означает, что шрифт внутри тега **** будет на одну градацию меньше, чем средний.

Гарнитурой шрифта называется внешний вид символов составляющих шрифтовой набор. Наиболее часто употребляемые гарнитуры: **Arial, Times New Roman, Courier New, MS Sans Serif**. Если на Web-странице присутствует тег:

```
<FONT FACE="COURIER NEW"> ...</FONT>
```

это означает, что весь текст, заключенный внутри тега, будет отображаться на экране шрифтом **Courier New**.

Цвет шрифта может задаваться с помощью параметра **COLOR** как в виде шестнадцатеричного числового кода, так и путем указания названия цвета (в языке HTML используются названия цветов, заимствованные из английского языка). Ниже перечислены названия цветов, которые чаще всего используются при оформлении Web-страниц, и дается перевод этих названий на русский язык:

AQUA – морской волны;
AZURE – лазурный;
BLACK – черный;
BLUE – синий;
BROWN – коричневый;
FIREBRICK – кирпичный;
FUCHSIA – лиловый;
GOLD – золотистый;
GRAY – серый;
GREEN – зеленый;
IVORY – слоновой кости;
KHAKI – хаки;
LIME – лайм;
MAROON – бордовый;
NAVY – темно-синий;
OLIVE – оливковый;
ORANGE – оранжевый;
PINK – розовый;
PLUM – сливовый;
PURPLE – пурпурный;
RED – красный;
SALMON – лососевый;
SILVER – серебристый;
SKYBLUE – небесно-голубой;
TAN – желтовато-коричневый;
TEAL – сине-зеленый;
VIOLET – фиолетовый;
WHEAT – пшеничный;
WHITE – белый;
YELLOW – желтый.

Если необходимо задать определенные параметры шрифта сразу во всем документе, то используют **одиночный** тег **<BASEFONT>**. Этот тег использует те же атрибуты, что и тег ****, но задаваемый им размер шрифта, цвет, гарнитура принимаются по умолчанию всем документом. В то же время, если в теге документа, оформленного тегом **<BASEFONT>**,

встречается фрагмент, заключенный в парный тег ****, данный тег получает приоритет над **<BASEFONT>**, т.е. указанный фрагмент оформляется в соответствии с атрибутами тега ****. Следует также учитывать, что тег **BASEFONT** «понимает» Internet Explorer, но его могут не знать некоторые другие браузеры.

Помимо тегов **** и **<BASEFONT>** в HTML применяется еще ряд тегов, используемых для оформления шрифта Web-документа. Существуют теги, позволяющие использовать различные начертания одного и того же шрифта. Это тег ** ... **, задающий полужирное начертание шрифта, и тег **<I>...</I>**, создающий курсивное начертание. Совместное использование этих тегов дает начертание шрифта в виде полужирного курсива. Кроме того, в электронных документах часто применяется такой способ оформления фрагмента текста, как его подчеркивание. Подчеркивание выполняется с помощью тега **<U>...</U>**.

В языке HTML существует также возможность оформлять текст в соответствии с его смысловым содержанием. Для смыслового выделения текста в HTML используется целая группа парных тегов. Как правило, текст, заключенный в смысловые теги, выделяется с помощью особого начертания или гарнитуры шрифта. Но если в браузере по каким-либо причинам нет возможности использовать различные начертания и гарнитуры шрифта, то для выделения фрагментов текста, оформленных этими тегами, браузер использует другой способ.

Так, при использовании тега **<CITE>...</CITE>** текст, находящийся внутри этого тега, рассматривается как цитата. Как правило, такой текст браузер выводит курсивным шрифтом.

Для выделения фрагмента текста используется тег **...**. Текст, заключенный в этот тег, также обычно выделяется курсивом. Другой способ выделения – использование тега **...**. В этом случае фрагмент текста выделяется, как правило, полужирным шрифтом.

К этой же группе относится несколько тегов, применяющихся для оформления фрагментов компьютерных программ. Например, для оформления исходного текста компьютерной программы используется парный тег **<CODE>...</CODE>**. Тег **<KBD>...</KBD>** позволяет выделить текст, который вводится в компьютер с клавиатуры. Тег **<VAR>... </VAR>** позволяет оформить имена переменных или параметры команд. Как правило, для оформления «компьютерных» тегов браузеры применяют моноширинный шрифт. Кроме того, имена переменных обычно выводятся курсивом, а текст, вводимый с клавиатуры, – полужирным шрифтом.

На Web-страницах, содержащих информацию учебного и научно-технического характера, часто возникает необходимость разместить различные математические, физические, химические формулы. Большинство таких формул содержит надстрочные или подстрочные индексы. Для создания надстрочных индексов в HTML используется парный тег

^{...}. Для подстрочных индексов используется другой парный тег **_{...}**.

Ниже приводится исходный текст Web-страницы, в которой используются указанные выше средства оформления текста:

```
<HTML>
<HEAD>
<TITLE>
Использование шрифта в HTML
</TITLE>
</HEAD>
<BODY>
<H1> Основы работы с шрифтом в HTML </H1>
<HR SIZE="5" COLOR="BLUE">
<FONT SIZE="4">
В HTML имеются следующие возможности для оформления
шрифта:
<BR><BR>
1. Можно увеличивать или уменьшать размер шрифта:
<BR>
Обычный шрифт
</FONT>
<FONT SIZE="2"> Мелкий шрифт </FONT>
<FONT SIZE="5"> Крупный шрифт. </FONT>
<BR><BR>
<FONT SIZE="4">
2. Можно использовать различные начертания шрифта:
<BR>
<B> полужирный, </B>
<I> курсив, </I>
<B><I> полужирный курсив, </I></B>
<U> подчеркивание. </U>
<BR><BR>
3. При написании формул можно использовать
<BR>надстрочные и подстрочные индексы:<BR>
(a+b)<SUP>2</SUP>=a<SUP>2</SUP>+2ab+b<SUP>2</SUP>
H<SUB>2</SUB>SO<SUB>4</SUB>
C<SUB>2</SUB>H<SUB>5</SUB>OH
</FONT>
</BODY>
</HTML>
```

Внешний вид страницы, содержащей различные варианты оформления шрифта, при просмотре ее в браузере, показан на рис.1.5.

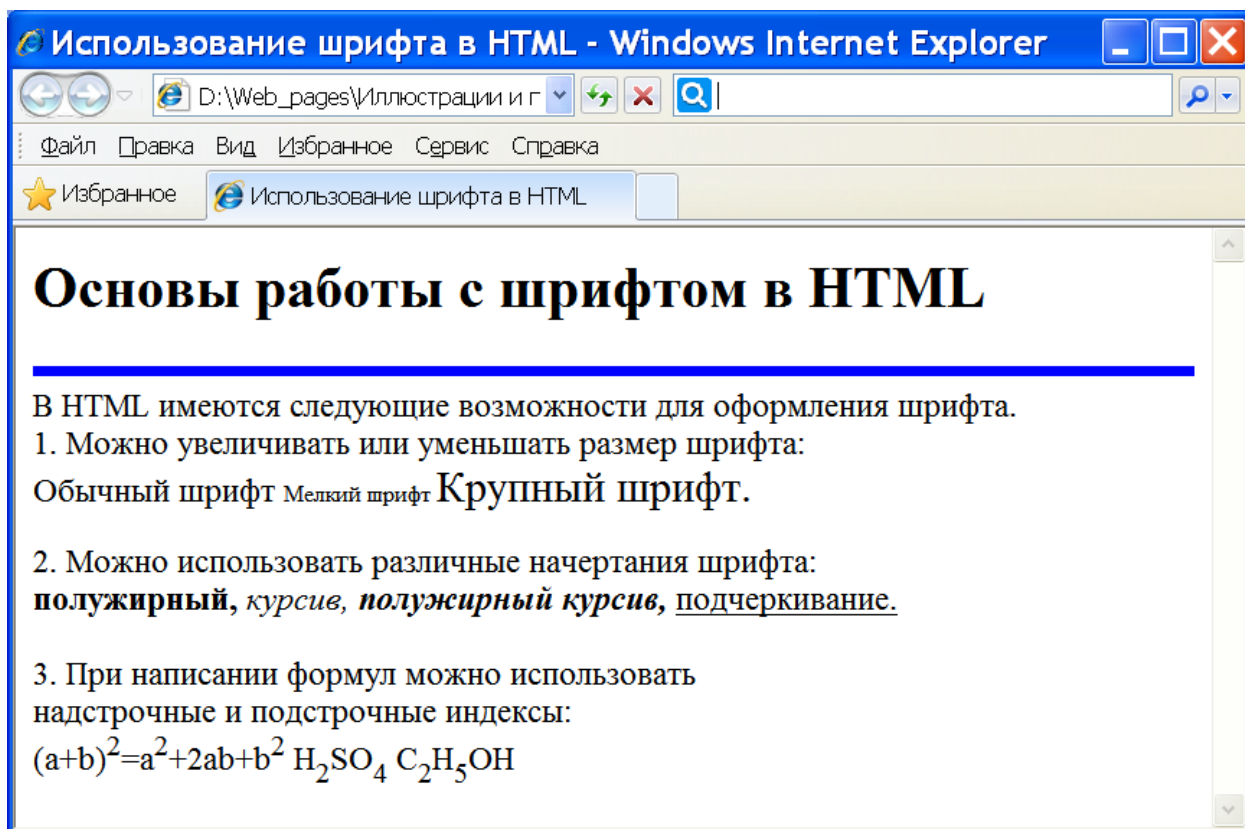


Рис.1.5. Внешний вид страницы, содержащей разные способы оформления шрифта, при просмотре ее в браузере Internet Explorer

Язык HTML позволяет изменять не только цвет шрифта, но и фоновый цвет самой Web-страницы. Для этого нужно использовать один из дополнительных атрибутов тега **<BODY>**. В открывающей части тега нужно указать параметр **BGCOLOR**. В качестве значения этого параметра, так же как и при указании цвета шрифта, можно указать название цвета или его шестнадцатеричный код. В последнем случае перед самым числовым кодом нужно поставить символ **#** (октоторп). Наличие октоторпа показывает, что идущую следом за ним числовую последовательность следует рассматривать именно как шестнадцатеричное, а не десятичное число. Например, светло-зеленый фоновый цвет для Web-страницы можно задать следующим тегом:

```
<BODY BGCOLOR="#CCFFCC">
```

Еще один дополнительный атрибут тега **<BODY>** позволяет задавать цвет текста во всем документе (аналогично тому, как это делает тег **<BASEFONT>**). Этот атрибут называется **TEXT**. Значением атрибута является цвет текста, который может быть записан в виде текстовой константы или с помощью шестнадцатеричного числового кода. Если тег **<BODY>** имеет следующие дополнительные атрибуты:

```
<BODY BGCOLOR="#CCFFCC" TEXT="BLUE"> ,
```

то Web-страница будет выводиться на светло-зеленом фоне, а текст этой страницы будет синего цвета. Преимуществом атрибута **TEXT** тега **<BODY>** по сравнению с тегом **<BASEFONT>** является его универсальность, поскольку его «знает» не только Internet Explorer, но и практически все другие браузеры.

1.4. Форматирование абзацев. Использование литералов

Оформление текста можно производить не только на уровне текстового фрагмента, но и на уровне абзаца. Создание нового абзаца в тексте документа производится с помощью парного тега **<P> ...</P>**. При создании нового абзаца текст автоматически начинается с новой строки и поэтому в данном случае использовать тег **
** не нужно. С помощью дополнительного атрибута **ALIGN** можно использовать разные способы выравнивания текста в абзаце. Атрибут **ALIGN** может принимать одно из 4 возможных значений:

- <P ALIGN="LEFT">** - выравнивание текста по левому краю;
- <P ALIGN="RIGHT">** - выравнивание текста по правому краю;
- <P ALIGN="CENTER">** - выравнивание текста по центру;
- <P ALIGN="JUSTIFY">** - выравнивание текста по ширине.

По умолчанию в HTML текст документа выравнивается по левому краю. В текстах Web-документа, написанных на русском языке, как правило, основной текст документа выравнивается по ширине, а заголовки различных уровней – по центру.

Для форматирования текста документа в языке HTML также используется парный тег **<DIV>...</DIV>**. С помощью этого тега в тексте страницы можно создать раздел более высокого уровня, который включает в себя несколько абзацев, т.е. в тег **<DIV>** могут быть вложены несколько тегов **<P>**.

Тег **<DIV>** обладает тем же дополнительным атрибутом **ALIGN**, что и тег **<P>**. Значения атрибута **ALIGN** тега **<DIV>** те же самые, и также позволяют выравнивать содержание соответствующего раздела по левому краю, центру, правому краю и ширине.

Еще одним широко используемым элементом оформления текста является использование красной строки в начале абзаца. Если в текстовом процессоре MS Word или другой подобной программе можно явно задать величину красной строки, то в HTML отсутствуют штатные средства, позволяющие это сделать.

Часто для того, чтобы отделить внутри Web-документа абзацы друг от друга, используют вставку пустой строки между абзацами. Но существует и другой способ выделения абзацев, позволяющий придать им такой же внешний вид, как и в документе, созданном в MS Word. Этот способ заключается в том, чтобы вставить в начало абзаца несколько пробелов.

Общее правило, принятое в HTML, заключается в том, что при обработке исходного текста документа учитывается только один пробел, вставленный между словами или в начале строки. Все последующие пробелы, сколько бы их ни было, HTML игнорирует. Но если задать специальным образом **неразрывный пробел**, то этот символ будет учитываться и отображаться соответствующим образом на Web-странице.

Неразрывный пробел в HTML нельзя непосредственно ввести с клавиатуры компьютера, но его можно указать в виде **литерала**. Литерал – это специальное обозначение, принятое для некоторых символов, которое начинается с символа **&** (амперсанд) и заканчивается точкой с запятой. Язык HTML понимает эти условные обозначения и вместо них вставляет в текст Web-документа соответствующие символы. Наиболее часто употребляются следующие литералы:

- <**; – знак «меньше»;
- >**; – знак «больше»;
- **; – неразрывный пробел;
- "**; – кавычки;
- &**; – амперсанд;
- ©**; – знак авторского права («копирайт»).

Таким образом, если в исходном тексте страницы, создаваемом в текстовом редакторе, 5 раз повторить литерал ** **; в начале абзаца, то при отображении этого текста на Web-странице, в начале абзаца появятся 5 пробелов, которые и создадут необходимую красную строку.

Еще одним случаем, когда без использования литералов не обойтись, является использование в тексте Web-документа символов «больше» и «меньше». Так как данные символы по начертанию совпадают с угловыми скобками, то по умолчанию они воспринимаются языком HTML как начало или конец какого-либо тега, что приведет к искажению вида и содержания страницы. Для того чтобы избежать такого недоразумения, вместо этих символов в исходном тексте документа вставляются литералы, и при просмотре Web-страницы в браузере там соответственно отображаются символы «больше» или «меньше».

Аналогичным образом выглядит ситуация и с двойными кавычками. Как известно, в двойные кавычки в HTML-коде заключаются значения атрибутов различных тегов. Поэтому, если в тексте документа встречается слово или фраза, заключенная в кавычки (не относящаяся к тегам), то браузер может принять ее за значение, присвоенное атрибуту какого-либо тега. Следовательно, в таком случае также лучше для обозначения кавычек использовать соответствующий литерал.

Ниже приводится исходный текст страницы, в которой задается светло-зеленый фоновый цвет, демонстрируются разные способы выравнивания текста в документе, а также создание красной строки с помощью неразрывных пробелов.

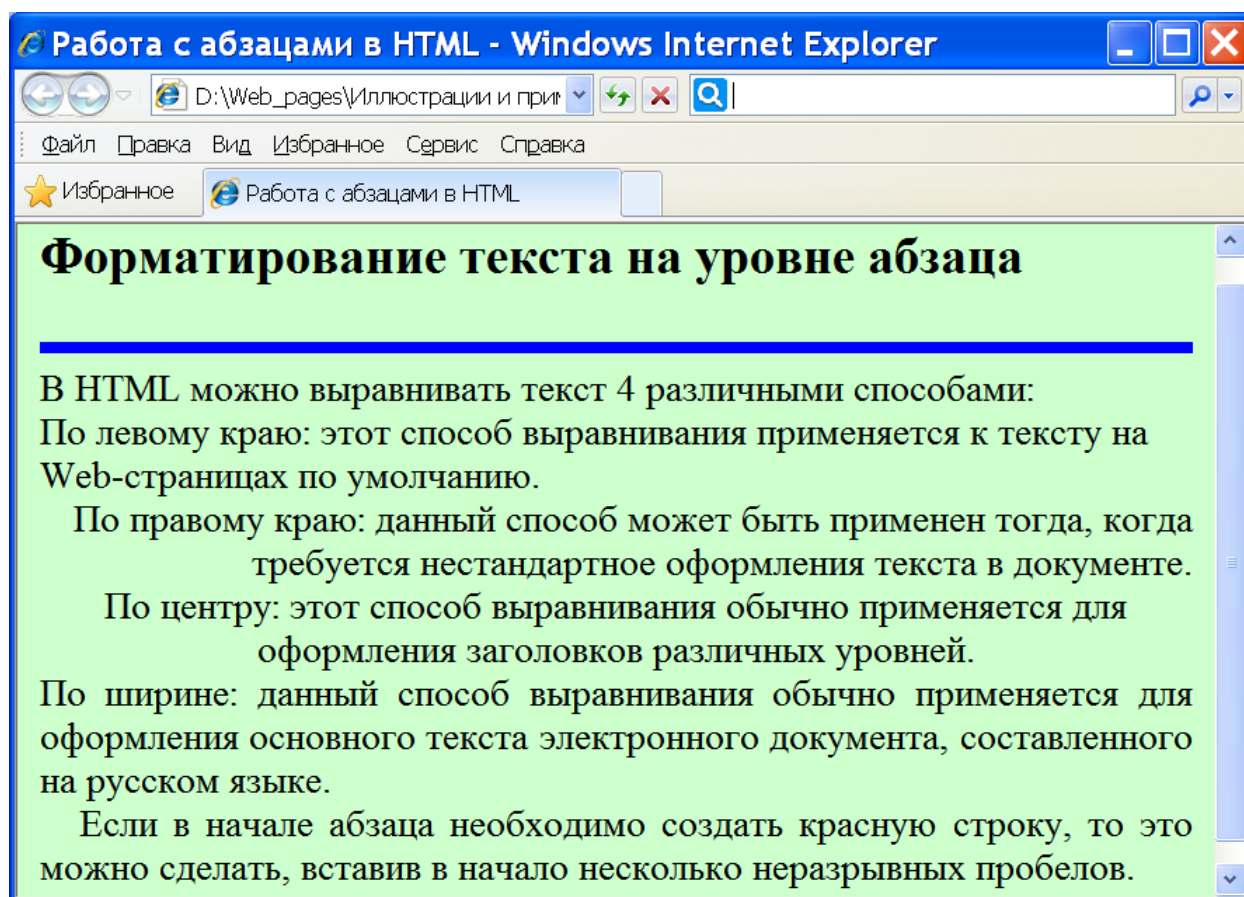


Рис. 1.6. Внешний вид страницы с 4 различными способами выравнивания текста и использованием красной строки в начале абзаца

Вопросы для самоконтроля

1. Как выглядит стандартная структура Web-страницы, созданной средствами языка HTML?
2. В каком формате нужно сохранять файл Web-страницы, созданной в текстовом редакторе?
3. Чем отличаются парные и одиночные теги?
4. Чем отличается открывающий и закрывающий теги?
5. Какой тег используется для задания заголовков различных уровней?
6. Какой тег используется для задания цвета, размера и гарнитуры текста?
7. Какие теги применяются для изменения начертания текста?
8. Какие теги применяются для создания нового параграфа в документе?
9. Как задать цвет текста для шрифта в числовой форме?
10. Как создать в тексте документа разделительную горизонтальную линию?
11. Как определить фоновый цвет для создаваемой Web-страницы?
12. Как создать красную строку в начале абзаца средствами HTML?
13. Как просмотреть исходный текст Web-страницы?
14. Как просмотреть содержимое Web-страницы в браузере, отличном от того, который является основным на данном компьютере?

ГЛАВА 2. Создание списков

Одним из наиболее распространенных способов структурирования и упорядочения информации, находящейся на Web-странице, является представление ее в виде списка. Существуют 3 вида списков, которые можно создать средствами HTML. Это нумерованные списки, маркированные списки и списки определений. В этой главе мы рассмотрим все виды списков.

2.1. Нумерованные списки

Нумерованным называется такой список, каждый элемент которого обозначается цифрой или буквой. Для создания нумерованного списка используется парный тег `<OL>...</OL>`. Отдельные элементы списка создаются с помощью непарного тега `<LI>`. По умолчанию, элементы списка нумеруются цифрами от 0 до 9. Для нумерации могут использоваться также римские цифры или латинские буквы.

Ниже приводится пример HTML-кода, который используется для создания простого нумерованного списка, содержащего перечень основных компонент современного компьютера:

```
<HTML>
<TITLE>
Обычный нумерованный список
</TITLE>
<BODY BGCOLOR="CCFFCC">
<H3> В состав персонального компьютера входят </H3>
<OL>
<LI> Системный блок
<LI> Монитор или дисплей
<LI> Клавиатура
<LI> Координатное устройство
<LI> Принтер
<LI> Сканер
<LI> Другие устройства
</OL>
</BODY>
</HTML>
```

Следует обратить внимание на то, что перед каждым элементом списка не надо указывать его порядковый номер, HTML делает это автоматически при правильном написании соответствующих тегов. Внешний вид данного списка при просмотре его в браузере показан на рис.2.1.

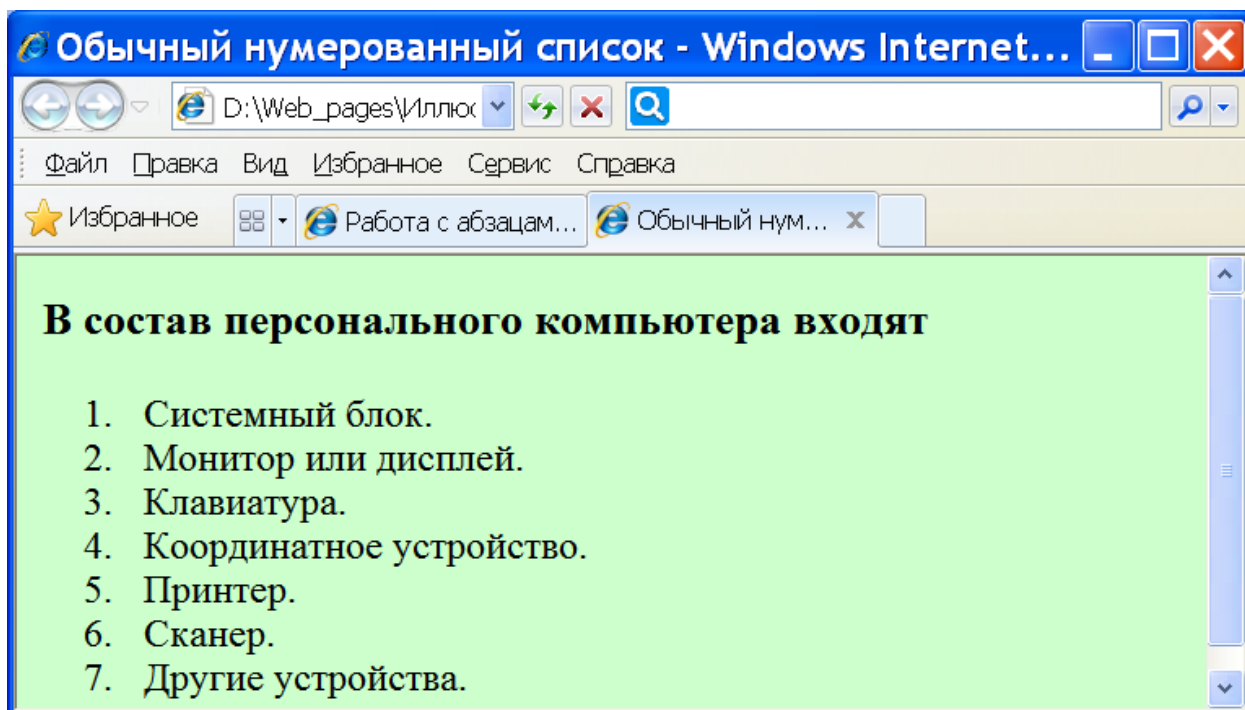


Рис. 2.1. Обычный нумерованный список

Списки обладают и еще одним свойством, которое видно на рис.2.1. Каждый элемент списка автоматически начинается с новой строки, поэтому для оформления списка не требуется для каждого его элемента ставить тег **
** или какой-либо другой, создающий в тексте новую строку.

Внимательно посмотрев на рис.2.1, можно отметить также следующее. В начале каждой строки – элемента списка имеется отступ от края текста. При этом в данном случае для создания такого отступа не использовалась вставка неразрывных пробелов, как это было сделано в предыдущей главе пособия. Отсюда можно сделать вывод, что красная строка в начале каждого элемента списка создается автоматически.

Как уже было указано, элементы нумерованного списка могут обозначаться и другими символами. Для этого необходимо в открывающейся части парного тега **...** указывать дополнительный атрибут **TYPE**. Например, в начале элементов списка могут стоять римские цифры, для чего атрибуту **TYPE** следует присвоить значение **I**:

```
<OL TYPE="I">
```

Ниже приводится HTML-код, который используется для создания страницы, содержащей список падежей, имеющих в русском языке. Все падежи пронумерованы римскими цифрами.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>
```

```
Нумерованный список с римскими цифрами
```

```
</TITLE>
```

```

</HEAD>
<BODY BGCOLOR="CCFFCC">
<H3> В русском языке существуют следующие падежи:
</H3>
<OL TYPE="I">
<LI> именительный;
<LI> родительный;
<LI> дательный;
<LI> винительный;
<LI> творительный;
<LI> предложный.
</OL>
</BODY>
</HTML>

```

Внешний вид списка с элементами, обозначенными римскими цифрами, приведен на рис.2.2.

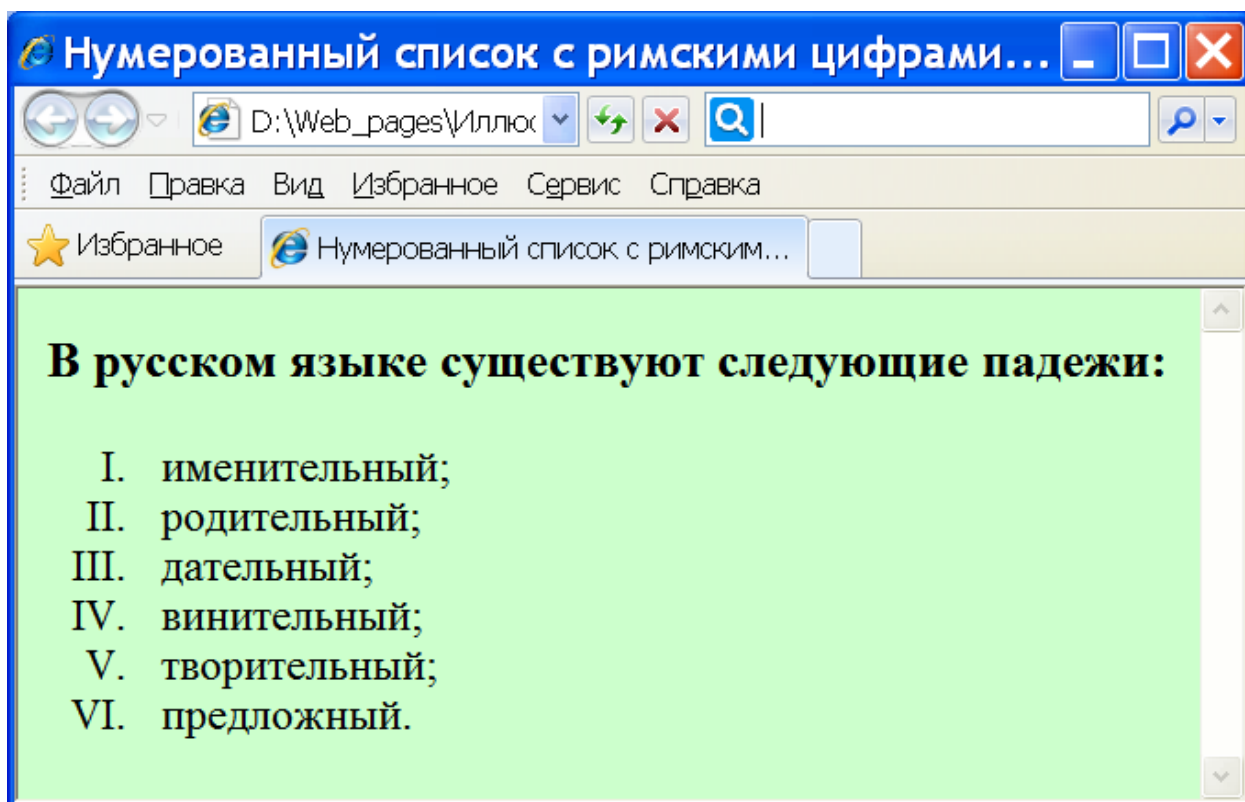


Рис. 2.2. Нумерованный список, в котором используются римские цифры

Еще одним вариантом оформления нумерованного списка является использование для обозначения элементов списка букв латинского алфавита. Для этого в теге **** атрибуту **TYPE** следует присвоить значение латинской буквы **A**:

<OL TYPE="A">

Рассмотрим данный вариант на следующем примере: необходимо создать перечень, основных устройств, находящихся на материнской плате персонального компьютера, причем каждый пункт перечня обозначить латинской буквой. Для выполнения поставленной задачи создадим Web-страницу с приведенным ниже кодом:

```
<HTML>  
<HEAD>  
<TITLE>  
Нумерованный список с латинскими буквами  
</TITLE>  
</HEAD>  
<BODY BGCOLOR="#CCFFCC">  
<H3> Материнская плата персонального компьютера  
содержит следующие основные устройства: </H3>  
<OL TYPE="A">  
<LI> центральный процессор, включающий в свой  
состав кэш-память;  
<LI> слот для установки центрального процессора;  
<LI> оперативное запоминающее устройство (ОЗУ);  
<LI> постоянное запоминающее устройство (ПЗУ);  
<LI> постоянное перепрограммируемое запоминающее  
устройство (ППЗУ);  
<LI> аккумулятор или батарейка для питания ППЗУ;  
<LI> генератор тактовых импульсов;  
<LI> таймеры;  
<LI> северный мост (управляет взаимосвязью  
процессора, оперативной памяти и видеоадаптера);  
<LI> южный мост, называемый также функциональным  
контроллером (выполняет функции контроллера жестких  
и гибких дисков, клавиатуры, координатного  
устройства, шины USB);  
<LI> шинные интерфейсы;  
<LI> интегрированный видеоадаптер;  
<LI> интегрированная аудиокарта;  
<LI> системный динамик;  
<LI> другие устройства.  
</OL>  
</BODY>  
</HTML>
```

Ниже приводится внешний вид страницы, содержащей нумерованный список с элементами, обозначенными латинскими буквами (рис.2.3):

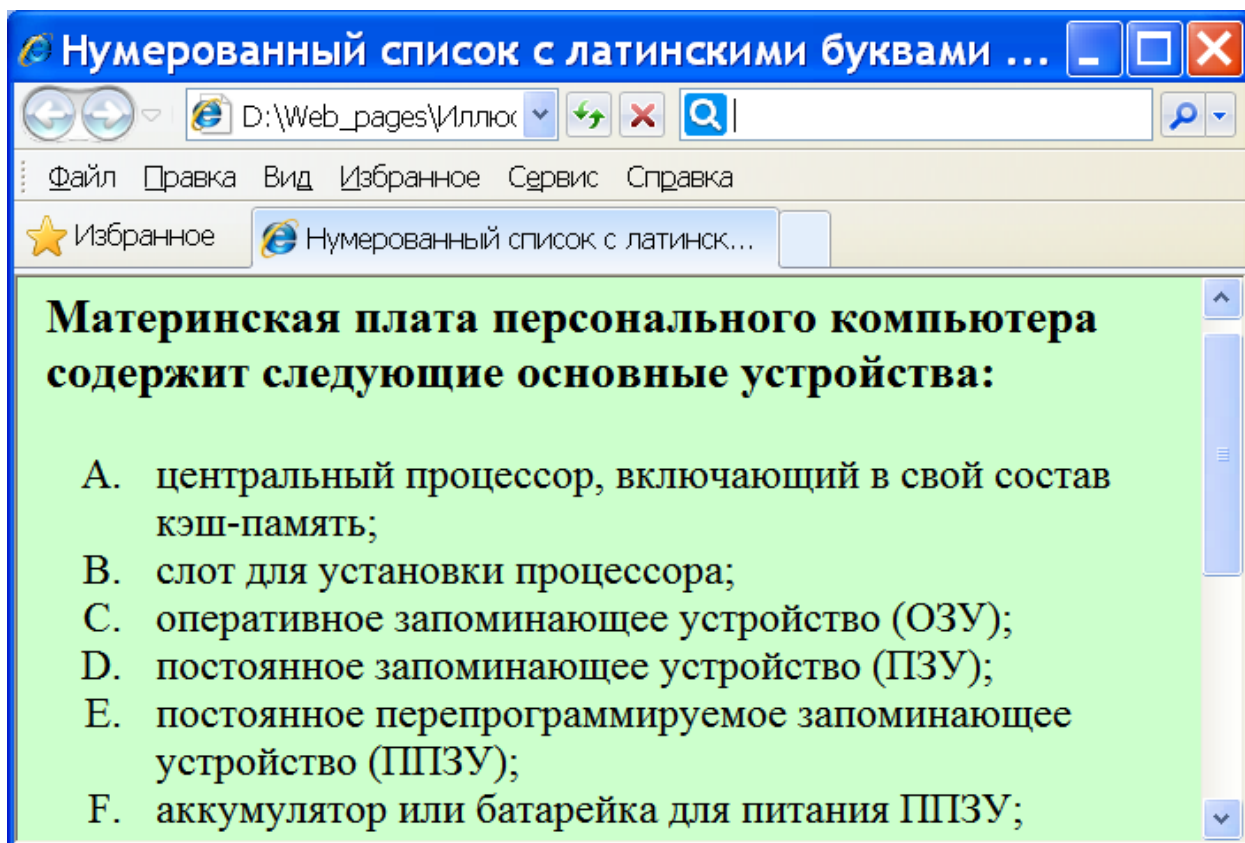


Рис. 2.3. Нумерованный список, элементы которого обозначаются прописными латинскими буквами

2.2. Маркированные списки

При разработке Web-страниц нередко возникает необходимость создать списки, в которых отдельные элементы обозначаются не буквами или цифрами, а специальными значками, которые называются маркерами. Данные списки называются маркированными. Маркированный список создается с помощью парного тега `<UL>...</UL>`. Очередной элемент списка создается тем же одиночным тегом `<LI>`, что и в случае с нумерованным списком.

Значок, с которого начинается очередной элемент списка, имеет одинаковую форму для всех элементов, относящихся к одному списку. По умолчанию формой значка-маркера является сплошной кружок.

Рассмотрим создание маркированного списка на примере перечня знаменитых архитекторов, создававших классический облик Санкт-Петербурга. Ниже приводится HTML-код соответствующей Web-страницы.

```
<HTML>
<HEAD>
<TITLE>
Обычный маркированный список
</TITLE>
```

```

</HEAD>
<BODY BGCOLOR="#CCFFCC">
  <H3> В XVIII - XIX веках в Санкт-Петербурге и его
пригородах жили и работали выдающиеся архитекторы:
</H3>
  <UL>
    <LI> Доменико Трезини
    <LI> Жан-Батист Леблон
    <LI> Франческо Бартоломео Растрелли
    <LI> Савва Чевакинский
    <LI> Антонио Ринальди
    <LI> Жан-Батист Вален-Деламот
    <LI> Этьен Фальконе
    <LI> Юрий Фельтен
    <LI> Джакомо Кваренги
    <LI> Василий Баженов
    <LI> Винченцо Бренна
    <LI> Иван Старов
    <LI> Чарльз Камерон
    <LI> Тома де Томон
    <LI> Андрей Воронихин
    <LI> Луиджи Руска
    <LI> Андреян Захаров
    <LI> Василий Стасов
    <LI> Карл Росси
    <LI> Огюст Монферран
    <LI> Константин Тон
    <LI> Андрей Штакеншнейдер
    <LI> Петр Клодт
    <LI> Александр Брюллов
  </UL>
</BODY>
</HTML>

```

Внешний вид страницы, содержащий маркированный список, показан на рис.2.4. Обратите внимание на то, что каждый элемент маркированного списка (при условии правильного написания соответствующих тегов) начинается с новой строки, так же, как и у нумерованных списков.

Если необходимо изменить внешний вид маркера, то следует использовать атрибут **TYPE** для парного тега ****. Например, вместо полностью закрашенных кружков в качестве маркеров можно использовать кружки, у которых обозначен только контур, а внутренняя их часть остается пустой. Для этого в открывающей части тега нужно использовать команду:

```
<UL TYPE="CIRCLE">
```

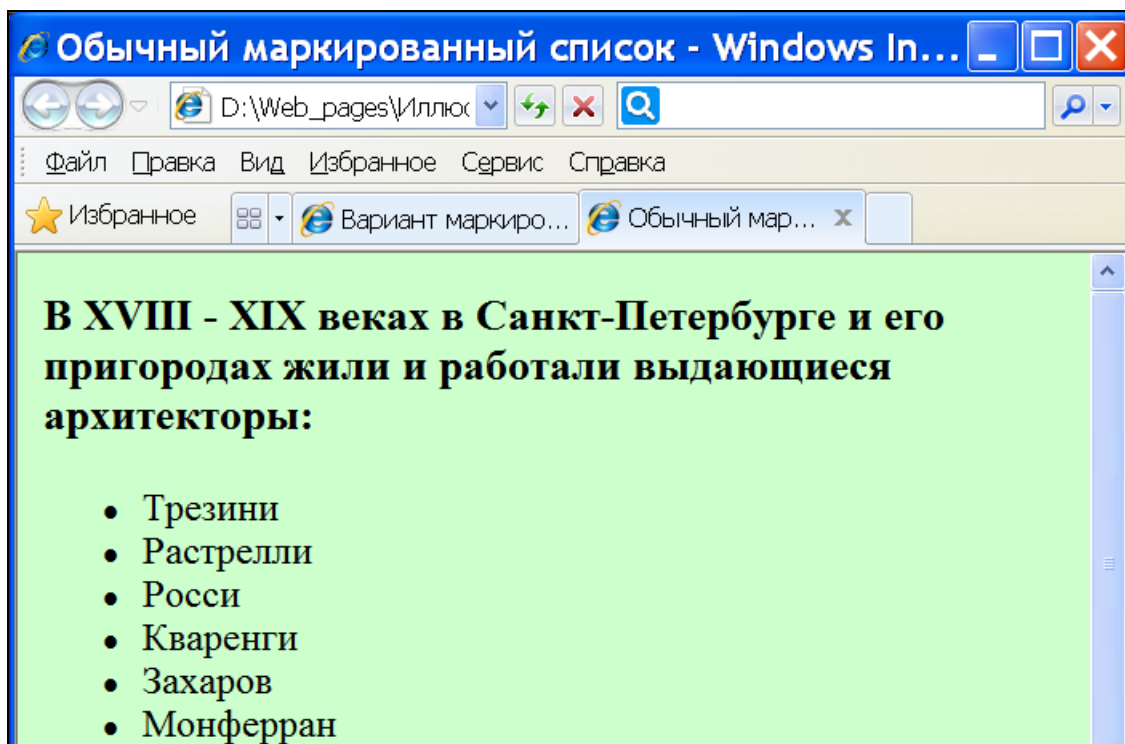


Рис. 2.4. Обычный маркированный список

Ниже приводится пример HTML-кода для создания Web-страницы с перечнем имеющихся на Земле океанов, в котором каждый элемент перечня обозначается кольцом с незаполненной полостью.

```
<HTML>
<HEAD>
<TITLE>
Вариант маркированного списка
</TITLE>
</HEAD>
<BODY BGCOLOR="CCFFCC">
<H3> На планете Земля существуют следующие
океаны:</H3>
<UL TYPE="CIRCLE">
<LI> Атлантический
<LI> Индийский
<LI> Тихий
<LI> Северный Ледовитый
</UL>
</BODY>
</HTML>
```

Далее на рис.2.5 демонстрируется внешний вид соответствующей Web-страницы.

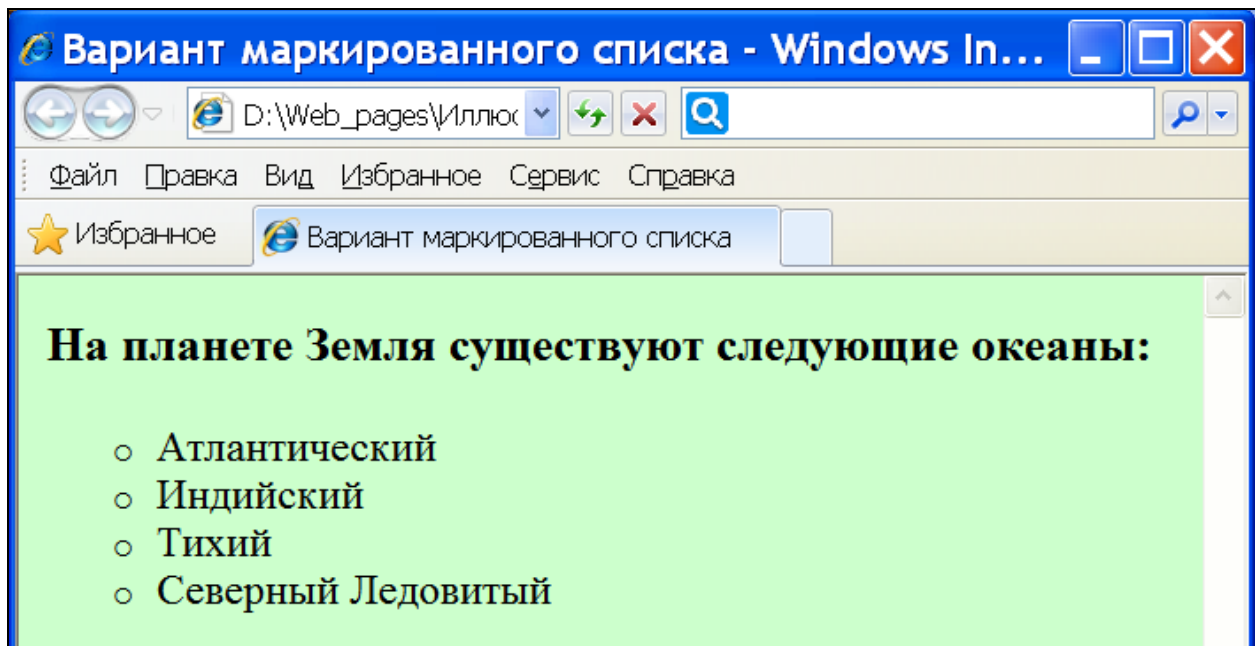


Рис. 2.5. Вариант маркированного списка с маркерами-кольцами

Маркеры в списке также могут иметь квадратную форму. Для этого значение атрибута **TYPE** необходимо изменить следующим образом:

<UL TYPE="SQUARE">

В качестве примера приведем Web-страницу, содержащую список частей света. Ниже представлен HTML-код данной страницы:

```
<HTML>
<HEAD>
<TITLE>
Вариант маркированного списка с квадратными
маркерами
</TITLE>
</HEAD>
<BODY BGCOLOR="CCFFCC">
<H3> На Земле существуют следующие части
света:</H3>
<UL TYPE="SQUARE">
<LI> Европа
<LI> Азия
<LI> Африка
<LI> Америка
<LI> Австралия и Океания
<LI> Антарктида
</UL>
</BODY>
</HTML>
```

На рис. 2.6 приведен внешний вид данной страницы при просмотре ее в браузере:

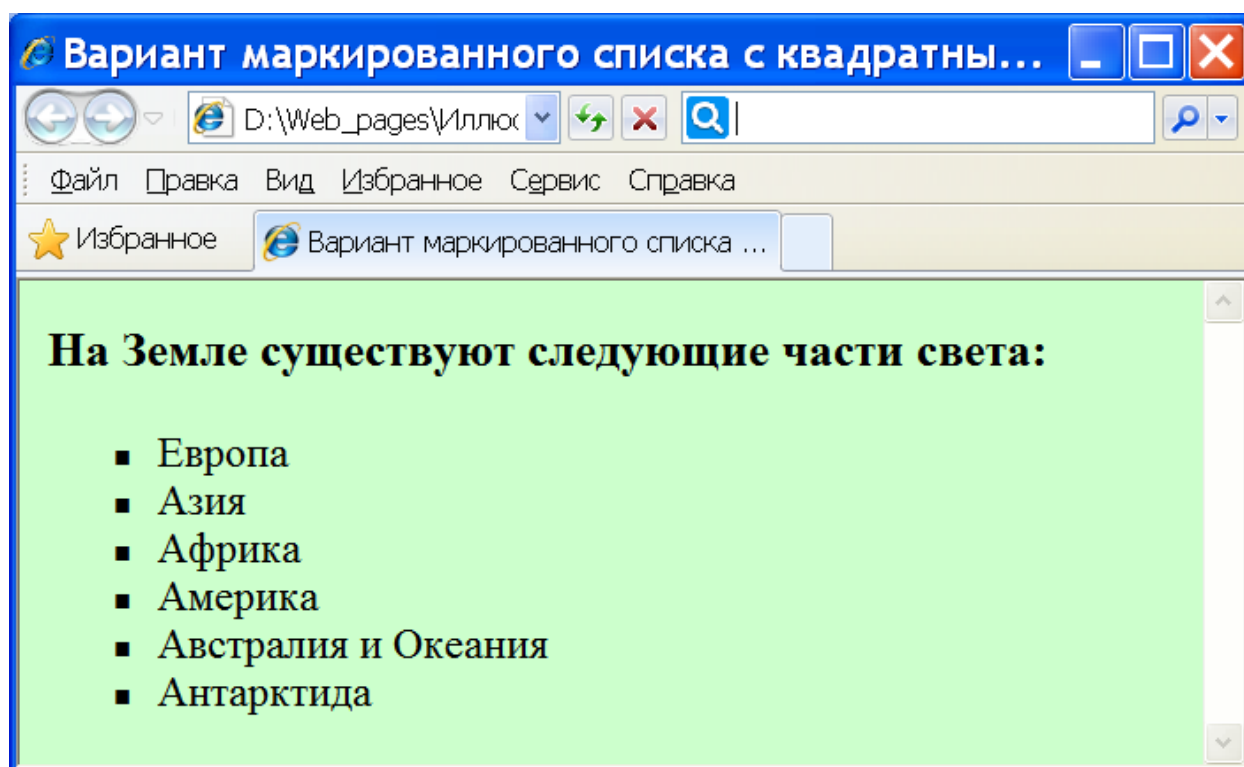


Рис. 2.6. Вариант маркированного списка с квадратными маркерами

2.3. Списки определений

Данный вид списка применяется для разработки Web-страниц, содержащих информацию справочного характера. Список определений может быть использован для представления на Web-странице справочника, словаря, глоссария. Такой список создается парным тегом **<DL>...</DL>**.

Отдельные элементы списка создаются одиночными тегами, но эти теги могут быть двух видов. Одиночный тег первого вида **<DT>**. Этот тег используется для добавления в список нового термина. Одиночный тег второго вида **<DD>**, как правило, следует за тегом **<DT>**. Этот тег содержит определение к приведенному выше термину. Общий вид списка определений выглядит следующим образом:

```
<DL>
<DT>
<DD>
<DT>
<DD>
.....
</DL>
```

Рассмотрим создание списка определений на следующем примере. Требуется создать перечень основных свойств алгоритма и для каждого из перечисленных свойств дать его определение. Данную задачу можно решить с помощью следующего HTML-кода:

```
<HTML>
<TITLE>
Список определений
</TITLE>
<BODY BGCOLOR="#CCFFCC">
<H3> Основные свойства алгоритма</H3>
<DL>
<DT> Дискретность
<DD>
<DIV ALIGN="JUSTIFY">
Алгоритм должен представлять процесс решения задачи
как последовательное выполнение простых шагов.
Каждое действие, предусмотренное алгоритмом,
исполняется только после того, как закончилось
исполнение предыдущего.
</DIV>
<DT> Определенность
<DD>
<DIV ALIGN="JUSTIFY">
Каждое правило алгоритма должно быть четким,
однозначным и не оставлять места для произвольного
толкования. Благодаря этому свойству выполнение
алгоритма носит механический характер и не требует
никаких дополнительных указаний или сведений о
решаемой задаче.
</DIV>
<DT> Результативность
<DD>
<DIV ALIGN="JUSTIFY">
Алгоритм должен приводить к решению поставленной
задачи за конечное число шагов
</DIV>
<DT> Массовость
<DD>
<DIV ALIGN="JUSTIFY">
Алгоритм решения задачи разрабатывается в общем
виде, то есть, он должен быть применим для
некоторого класса задач, различающихся только
исходными данными.
</DIV>
```

```
</DL>
</BODY>
</HTML>
```

Внешний вид получившейся в результате Web-страницы приведен ниже на рис.2.7.

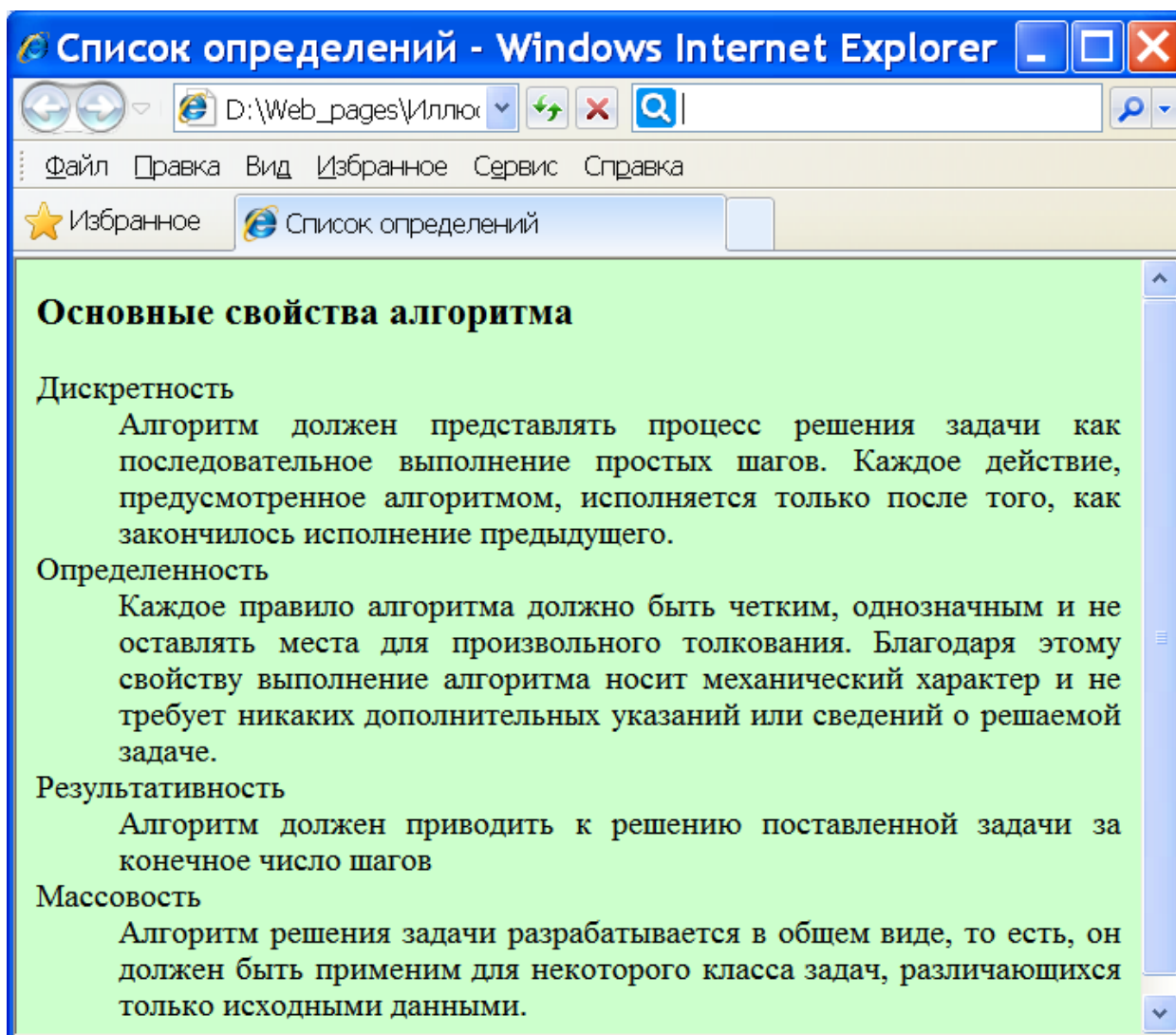


Рис. 2.7. Список определений, содержащий перечень основных свойств алгоритма

2.4. Многоуровневые списки

Язык HTML позволяет создавать списки, в котором любой из элементов списка может содержать список более низкого уровня. Такой список, содержащий два или более уровня списков, называется многоуровневым. Структура списков второго и последующих уровней такая же, как и у списка первого уровня, т.е. сам список создается с помощью соответствующего парного тега, а отдельные элементы списка – одиночными тегами. К тегам, создающим список, в случае необходимости добавляются дополнительные атрибуты.

Примером такого списка является перечень планет Солнечной системы. Создадим Web-страницу, которая содержит список из 8 больших планет, имеющих в Солнечной системе. Этот перечень должен содержать не только названия планет, но и названия спутников, которые есть у всех планет, кроме Марса и Меркурия. Для планет-гигантов, начиная с Юпитера, будем указывать только крупнейшие спутники. Сами планеты (элементы списка первого уровня) обозначены римскими цифрами, а спутники планет (элементы списка второго уровня) обозначены квадратными маркерами.

```
<HTML>  
<TITLE>  
Многоуровневый список  
</TITLE>  
<BODY BGCOLOR="CCFFCC">  
<H3>Солнечная система</H3>  
<DIV ALIGN="JUSTIFY">  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
На этой странице содержится список планет Солнечной системы и их спутников. Планеты обозначены римскими цифрами, а спутники - квадратными маркерами. Для планет-гигантов указаны только их крупнейшие спутники.  
</DIV>  
<HR SIZE=5 COLOR=BROWN>  
<BR>  
<OL TYPE="I">  
<LI> Меркурий  
<LI> Венера  
<LI> Земля  
    <UL TYPE=SQUARE>  
        <LI> Луна  
    </UL>  
<LI> Марс  
    <UL TYPE=SQUARE>  
        <LI> Фобос  
        <LI> Деймос  
    </UL>  
<LI> Юпитер  
    <UL TYPE=SQUARE>  
        <LI> Ганимед
```

```

    <LI> Европа
    <LI> Ио
    <LI> Каллисто
  </UL>
<LI> Сатурн
  <UL TYPE=SQUARE>
    <LI> Титан
    <LI> Япет
    <LI> Тефия
    <LI> Рея
    <LI> Диона
  </UL>
<LI> Уран
  <UL TYPE=SQUARE>
    <LI> Титания
    <LI> Оберон
    <LI> Ариэль
    <LI> Умбриэль
  </UL>
<LI> Нептун
  <UL TYPE=SQUARE>
    <LI> Тритон
    <LI> Нереида
  </UL>
</OL>
</BODY>
</HTML>

```

Ниже, на рис.2.8, показан внешний вид многоуровневого списка. Просматривая эту иллюстрацию, следует обратить внимание на то, как выглядят на данной странице отступы в элементах списка.

В строках с элементами списка первого уровня (обозначенных римскими цифрами) имеется отступ по сравнению с основным текстом, расположенным вверху страницы. А в строках с элементами списка второго уровня (обозначенных квадратными маркерами) имеется дополнительный отступ по сравнению с элементами списка первого уровня. Таким образом, в строках с элементами списка второго уровня создается отступ двойной величины по сравнению со стандартным.

Следовательно, чем ниже уровень элемента списка, тем правее он сдвигается по отношению к основному тексту Web-страницы. Поэтому уже по внешнему виду элемента можно определить его уровень в иерархической структуре списка. При этом все указанные сдвиги текста теги многоуровневого списка создают автоматически, и нет необходимости регулировать величину отступа с помощью вставки пробелов, как это мы ранее делали в абзацах обычного текста, не входящего в списки.

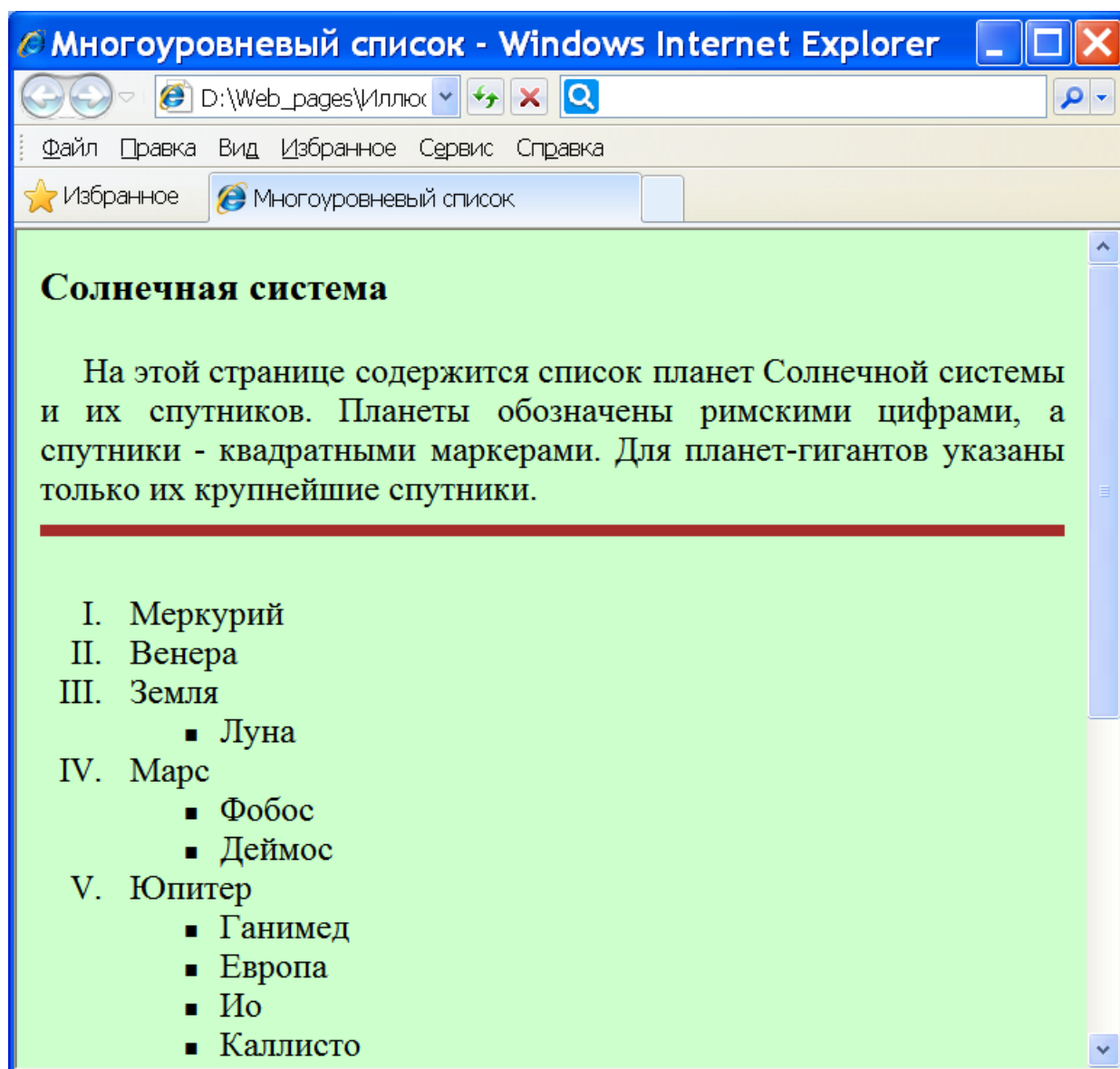


Рис. 2.8. Многоуровневый список, содержащий перечень планет Солнечной системы и их спутников

Вопросы для самоконтроля

1. Перечислите основные виды списков, применяемые на Web-страницах?
2. Какой парный тег применяется для создания маркированного списка?
3. Какой парный тег применяется для создания нумерованного списка?
4. Какой одиночный тег применяется для создания нового элемента нумерованного или маркированного списка?
5. Какой парный тег применяется для создания списка определений?
6. Какие одиночные теги используются в списках определений?
7. Каким образом можно создать многоуровневый список на Web-странице?
8. Какие атрибуты применяются для изменения внешнего вида маркеров в маркированном списке?
9. Какие атрибуты применяются для того чтобы в нумерованном списке отдельные пункты обозначались римскими цифрами?
10. Какие атрибуты применяются для того чтобы в нумерованном списке отдельные пункты обозначались латинскими буквами?
11. Чем отличается список определений от нумерованного или маркированного списка?
12. Какие виды списков могут использоваться в многоуровневом списке?
13. Какие виды маркеров могут использоваться в маркированном списке?
14. Могут ли элементы многоуровневого списка относиться к разным типам?

ГЛАВА 3. Использование графических элементов

В настоящее время Web-страницы только в редких случаях содержат исключительно один текст. Как правило, информация, находящаяся на Web-странице, представляет собой сочетание текста и различных элементов компьютерной графики.

Отличие использования компьютерной графики от текстовой информации заключается в следующем. Текст, содержащийся на странице, записывается непосредственно в исходном коде этой страницы и затем оформляется с помощью тегов. Графические элементы содержатся в отдельных файлах и с помощью специального тега подключаются к основному содержимому страницы. Файлы с компьютерной графикой могут находиться в той же папке, что и сама Web-страница, могут находиться в другой папке того же сайта или вообще на другом сайте.

3.1. Основные форматы графических файлов

Существует большое количество форматов графических файлов, но не все из них подходят для представления графической информации на Web-страницах. Основным ограничителем является в данном случае размер файлов. Так как файлы большого размера могут долго загружаться, то их наличие на сайте может сильно тормозить работу пользователя во всемирной сети. Естественно, что пользователь быстро потеряет интерес к такому «громоздкому» сайту. Поэтому для облегчения загрузки сайта следует использовать те форматы, которые обеспечивают небольшой объем графических файлов при достаточно высоком качестве содержащихся в них изображений.

Лучше всего для данной цели подходят файлы форматов **gif** и **jpeg**, которые получили широкое распространение во всемирной сети и распознаются всеми современными браузерами.

Файлы формата **gif** имеют одноименное расширение **gif**. Название формата представляет собой аббревиатуру, которая расшифровывается как **graphic interchange format**, что, в переводе с английского языка, означает «формат для обмена графикой». Изображения в этом формате содержат 256 различных цветов.

Файлы формата **jpeg** имеют расширение **jpeg** или **jpg**. Название формата представляет собой сокращенное название организации, разработавшей этот формат (**Join Photographic Experts Group**). Этот формат изначально предназначался для хранения в компьютере фотографических изображений, но в настоящее время он широко используется и для других видов компьютерной графики. Формат **jpeg** использует 24-битный цвет, т.е. в изображении данного формата можно использовать более 16 миллионов различных цветов.

Если разработчик сайта хочет использовать на создаваемой им Web-странице иллюстрации, созданные в других форматах, то их следует открыть в каком-либо графическом редакторе, а затем заново сохранить в одном из двух вышеуказанных «компактных» форматов.

Это можно сделать, например, с помощью программы **Paint**, являющейся стандартным приложением операционной системы **Windows**. В данном графическом редакторе команда «**Сохранить как ...**» открывает диалоговое окно, в котором можно выбрать один из нескольких форматов для сохранения файла, в том числе форматы **gif** и **jpeg**.

В частности, такую операцию переформатирования графических файлов рекомендуется применять в том случае, если планируется использовать иллюстрацию, изначально имевшую стандартный для Windows формат **bmp**. При сохранении такой иллюстрации в формате **jpeg** объем файла уменьшается более чем на порядок, а потери качества изображения получаются минимальными.

3.2. Вставка иллюстраций в текст страницы

Для вставки изображения на Web-странице используется одиночный тег **IMG SRC**. Общий вид тега следующий:

```
<IMG SRC="picture.ext">
```

где **picture** – это имя графического файла, а **ext** – расширение имени.

Например, тег

```
<IMG SRC="kart1.jpg">
```

вставляет на текущей странице изображение, взятое из файла **kart1** с расширением **jpg**, находящегося в той же папке, что и сама Web-страница. Если графический файл находится в другой папке сайта (и тем более на другом сайте), то необходимо правильно указать путь к этой папке.

Если иллюстрация находится в папке, которая вложена в текущую папку, содержащую саму страницу (т.е. папка с иллюстрациями является по отношению к текущей папкой нижнего уровня), то тег для вставки изображения будет выглядеть таким образом:

```
<IMG SRC="pictures/kart1.jpg">
```

где **pictures** – это название вложенной папки, содержащей графические файлы, а слэш используется для того, чтобы отделить название папки от названия файла.

Тег **** обладает рядом дополнительных атрибутов, позволяющих регулировать размеры изображения, расстояние между изображением и текстом, взаимное расположение текста и изображения и выполнять некоторые другие задачи.

По умолчанию изображение на Web-странице выводится в натуральную величину, т.е. в том же размере, что и в исходном файле, содержащем это изображение. Но в случае необходимости можно определять горизонтальные и вертикальные размеры изображения с помощью атрибутов **WIDTH** (ширина) и **HEIGHT** (высота). Размеры изображения измеряются в пикселях. Например, тег следующего вида:

```
<IMG SRC="kart1.jpg" WIDTH="300" HEIGHT="200">
```

выделяет на Web-странице для иллюстрации, взятой из файла **kart1.jpg**, окно размером 300 пикселей по горизонтали и 200 пикселей по вертикали. Выделение места под иллюстрацию браузер производит еще до загрузки самой иллюстрации, что несколько ускоряет отображение страницы.

Если фактические размеры рисунка меньше, чем указанные с помощью атрибутов **WIDTH** и **HEIGHT**, то рисунок растягивается до размеров, заданных данными атрибутами. Если рисунок больше, чем отведенное для него окно, то он, соответственно, сжимается. В любом случае, вне зависимости от соотношения размеров окна и фактических размеров иллюстрации, на Web-странице отображается вся иллюстрация в целом, а не отдельный ее фрагмент.

По умолчанию изображение на Web-странице располагается вплотную к тому тексту, который соседствует с этим изображением, что выглядит не очень эстетично при просмотре страницы. Поэтому тег **** имеет дополнительные атрибуты, позволяющие задать расстояние по горизонтали и по вертикали между изображением и текстом. Эта задача выполняется с помощью атрибутов **HSPACE** (расстояние по горизонтали) и **VSPACE** (расстояние по вертикали). Эти величины также измеряются в пикселях. Пример использования данных атрибутов:

```
<IMG SRC="kart1.jpg" HSPACE="30" VSPACE="15">
```

В этом теге между иллюстрацией **kart1** и текстом задается промежуток в 30 пикселей по горизонтали и 15 пикселей по вертикали.

Ниже приводится HTML-код страницы, которая содержит информацию о развитии современной компьютерной техники. Помимо текста, страница содержит изображение персонального компьютера, загруженное из файла **comp.jpg**. Под иллюстрацию выделено окно размером 220 пикселей по горизонтали и 150 пикселей по вертикали. Между окном с изображением и текстом имеется промежуток в 20 пикселей по горизонтали и по вертикали. На рис. 3.1 показан внешний вид данного окна при его просмотре в браузере.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>
```

```
Использование графики на Web-страницах
```

```
</TITLE>
```


столе, а успешно работать на таком компьютере может даже школьник. Но и такой стационарный компьютер уже не является последним словом в развитии информационных технологий. Все большее распространение получают мобильные компьютерные устройства: ноутбуки, планшеты, смартфоны.

</DIV>

<BODY>

<HTML>

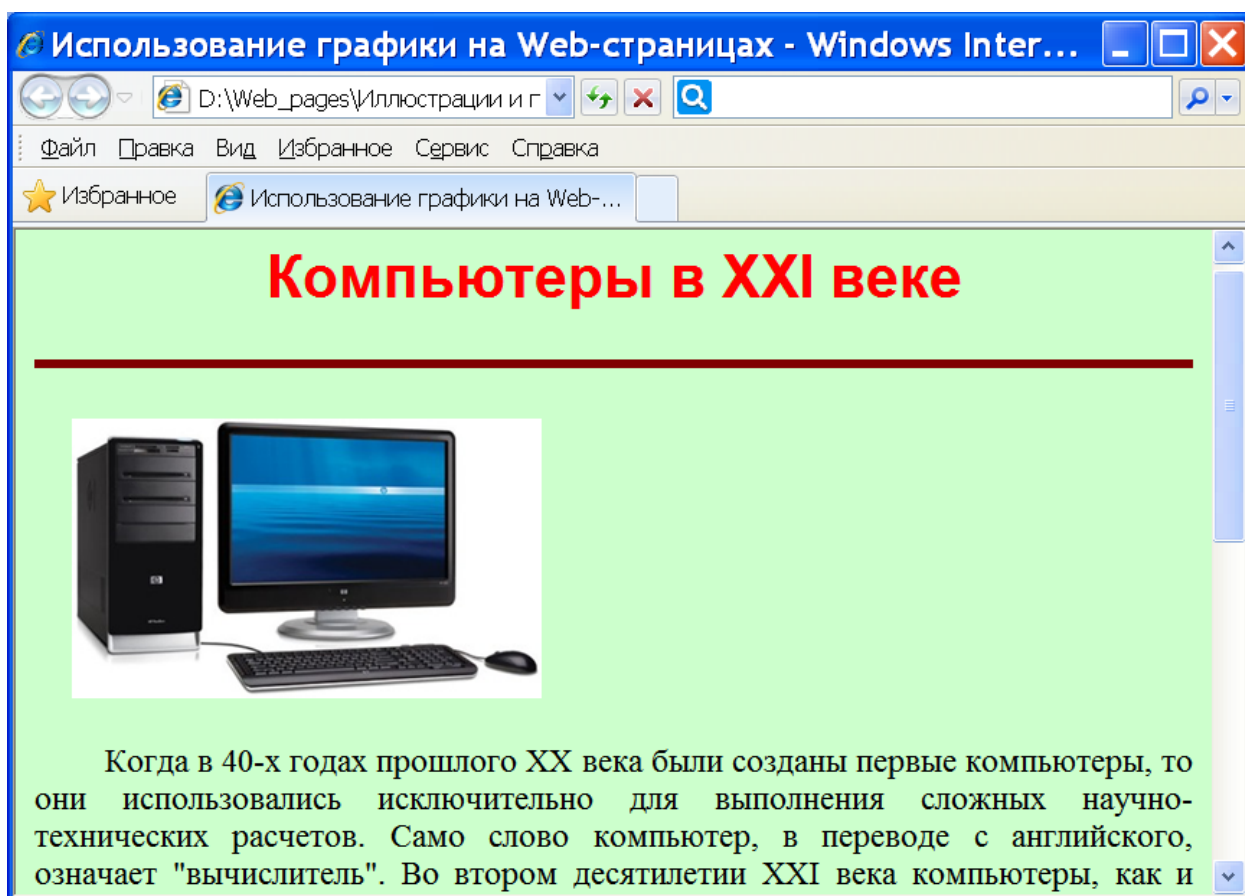


Рис. 3.1. Страница, содержащая иллюстрацию в формате jpeg

Если посмотреть на страницу, представленную на рис.3.1, то можно отметить следующий ее недостаток: в правой верхней части страницы имеется достаточно большое пустое пространство, не заполненное ни текстом, ни изображением. Естественно, что страница смотрелась бы гораздо лучше, если бы на месте этого «белого пятна» располагалась какая-либо полезная информация.

Для того чтобы устранить указанный недостаток, можно использовать атрибут **ALIGN** тега ****. Данный атрибут позволяет задать взаимное расположение изображения и текста. Атрибут **ALIGN** может принимать пять различных значений, но чаще всего используются два: **LEFT** и **RIGHT**. При

использовании атрибута **LEFT** изображение размещается у левого края страницы, а следующий за ним текст располагается справа от него. При задании атрибута **RIGHT** изображение помещается у правого края страницы, а последующий текст размещается слева.

Рассмотрим использование атрибута **ALIGN** на следующем примере. Возьмем страницу, показанную на рис.3.1. и внесем в нее одно небольшое изменение. В тег **IMG SRC** добавим атрибут **ALIGN** со значением **LEFT**. Тогда тег приобретет следующий вид:

```
<IMG SRC="comp.jpg" WIDTH="220" HEIGHT="150"
HSPACE="20" VSPACE="20" ALIGN="LEFT">
```

Сохраним изменения в HTML-коде и заново посмотрим, как выглядит данная страница при просмотре ее в браузере.

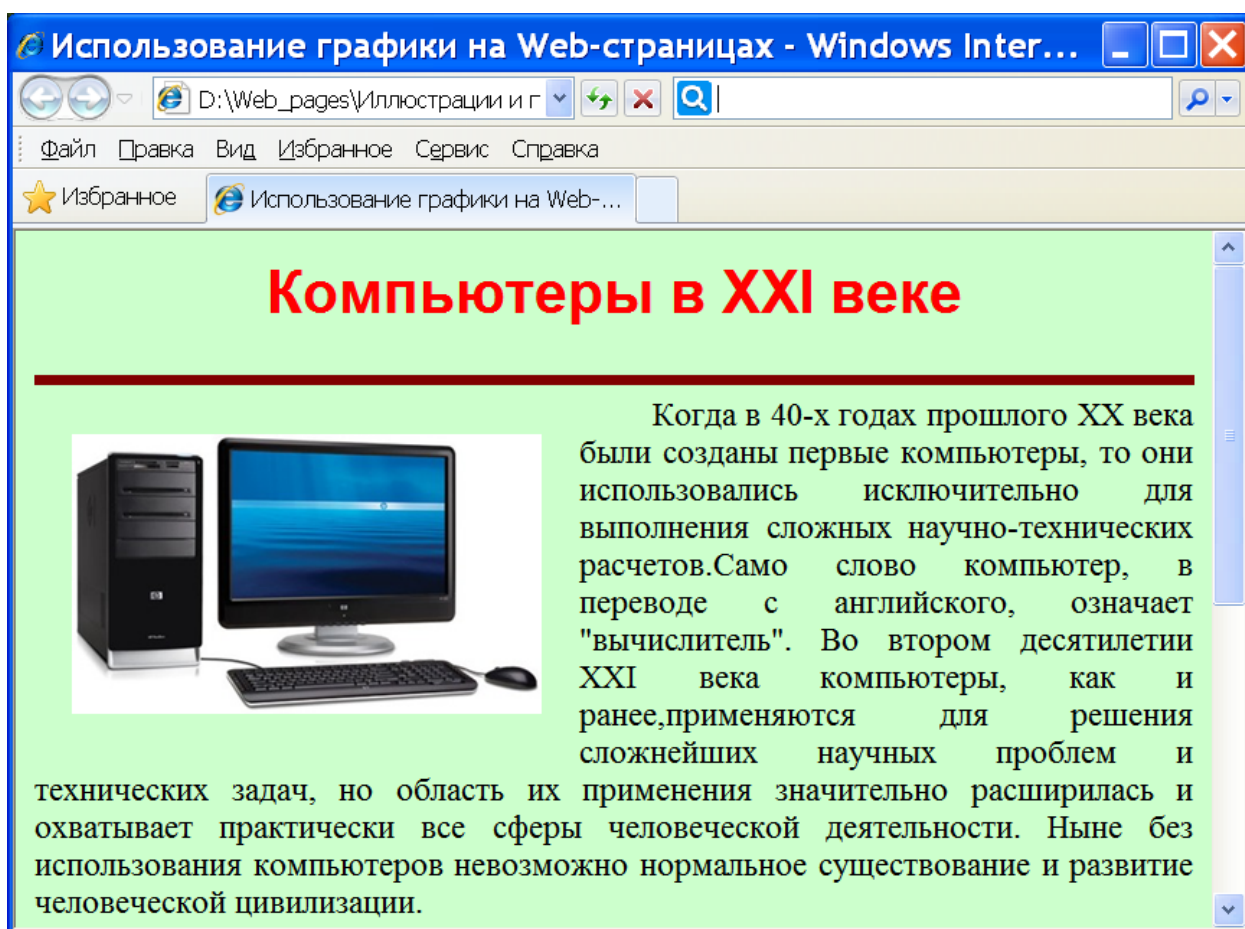


Рис. 3.2. Страница, содержащая изображение, которое помещается у левого края

Если посмотреть на обновленную Web-страницу с изображением (см. рис.3.2), то можно заметить, что «белое пятно» на этой странице исчезло, а его место занял текст, что улучшило внешний вид данной страницы.

Проведем еще один эксперимент с этой же страницей. В теге **** изменим значение атрибута **ALIGN** с **LEFT** на **RIGHT**. Тогда тег будет выглядеть следующим образом:

```
<IMG SRC="comp.jpg" WIDTH="220" HEIGHT="150" HSPACE="20" VSPACE="20" ALIGN="RIGHT">
```

Зафиксируем изменения в HTML-коде и обновим страницу. Внешний вид страницы после выполненных изменений представлен на рис.3.3.

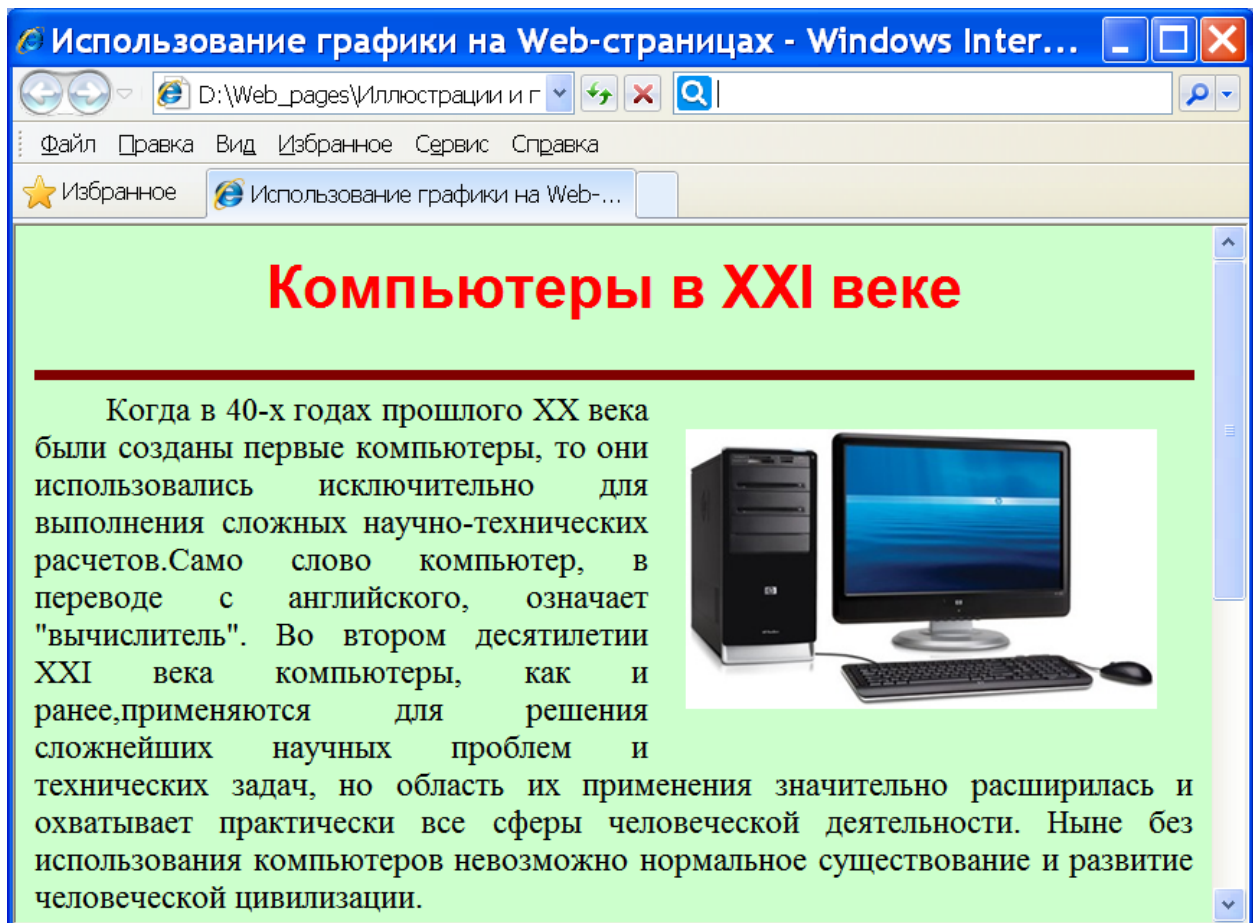


Рис.3.3. Страница, содержащая изображение, которое помещается у правого края

Как видно на рис.3.3, взаимное расположение графического элемента и текста изменилось на прямо противоположное. Но и в этом случае на странице не остается большого незаполненного пространства.

Существует еще три значения атрибута **ALIGN** тега ****:

- при значении **TOP** верхняя граница изображения выравняется по верхнему краю текста;
- при значении **MIDDLE** середина изображения совмещается с верхним краем текста;
- при значении **BOTTOM** нижняя граница изображения выравняется по нижнему краю текста.

Следует отметить, что использование этих значений, как правило, менее эффективно, чем значений **LEFT** и **RIGHT**. Поэтому значения атрибута **ALIGN**, выполняющие выравнивание текста и изображения по вертикали, применяются значительно реже.

Нередко возникает необходимость для лучшего оформления графического элемента создать вокруг него рамку. Для этого можно воспользоваться еще одним атрибутом тега ****. Это атрибут **BORDER**, значением которого является толщина рамки, выраженная в пикселях. Если в HTML-код рассмотренной ранее в этой главе страницы в тег **** добавить атрибут **BORDER**, то тег приобретет следующий вид:

```
<IMG SRC="comp.jpg" WIDTH="220" HEIGHT="150"
HSPACE="20" VSPACE="20" ALIGN="RIGHT" BORDER="5">
```

Внешний вид страницы после внесенных в тег изменений показан на рис.3.4. Теперь имеющаяся на странице иллюстрация обведена рамкой толщиной в 5 пикселей. Цвет рамки совпадает с цветом основного текста, имеющегося на странице. Поскольку основным цветом текста по умолчанию является черный, то и рамка черного цвета. Если бы основным цвет текста был синим, то и рамка была бы синего цвета и т.д.

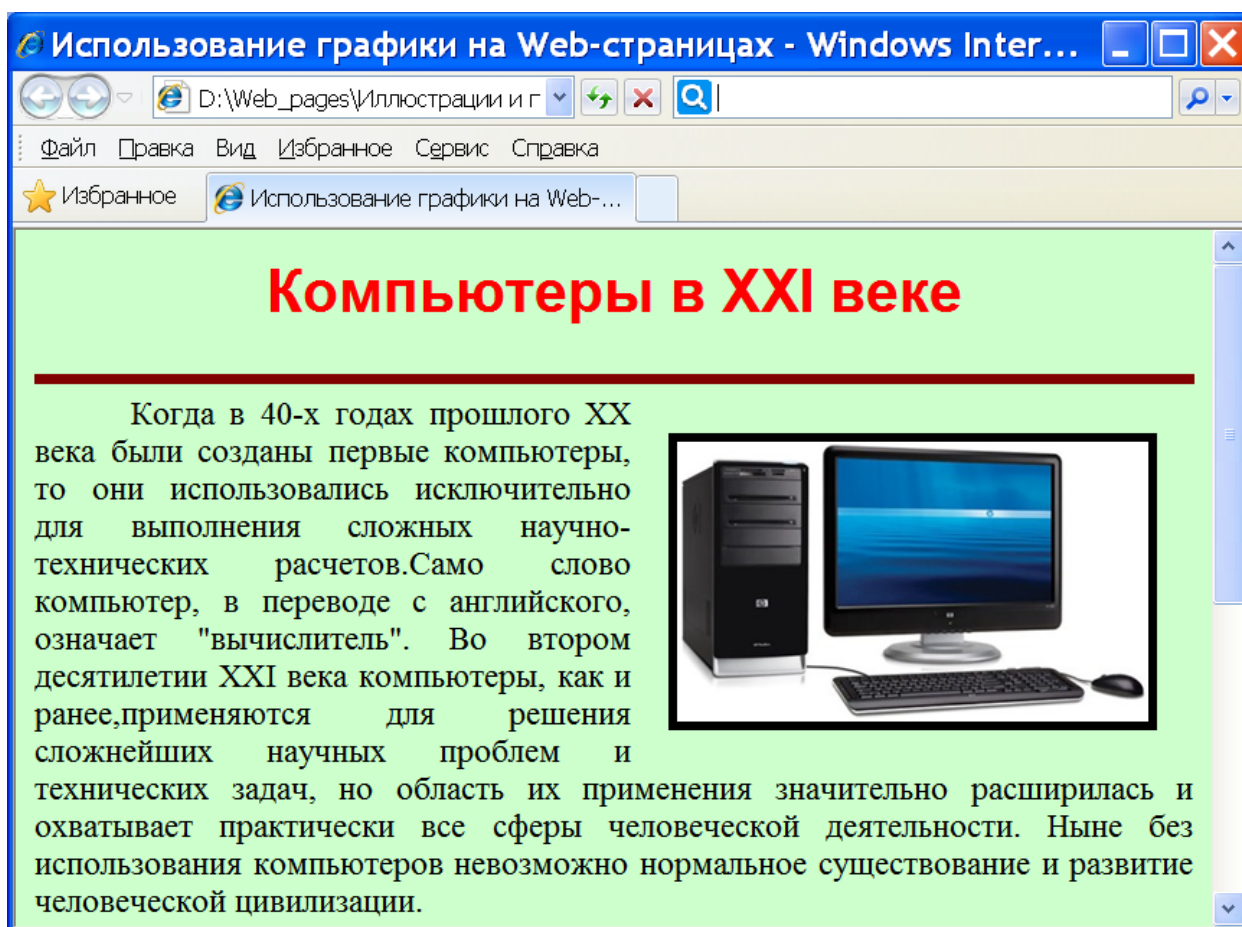


Рис.3.4. Страница с иллюстрацией, обведенной рамкой

Остановимся еще на одном атрибуте тега ****. Целесообразность использования этого атрибута обусловлена тем, что может возникнуть ситуация, когда текст документа нормально отображается в браузере, а иллюстрации по каким-либо причинам не загружаются. В этом случае желательно, чтобы вместо иллюстрации был не просто пустой прямоугольник, а в этом прямоугольнике находилась бы надпись, сообщающая о содержании данной иллюстрации.

Для этого в тег **** нужно добавить еще один атрибут **ALT**. Значением этого атрибута является содержание надписи, которая будет появляться на графическом объекте при наведении на него указателя мыши. Всплывающая надпись будет появляться на объекте вне зависимости от того, виден ли на странице сам объект, или пользователь может видеть только его контур. В нашем примере дополним тег **** еще одним атрибутом:

```
<IMG SRC="comp.jpg" WIDTH="220" HEIGHT="150"
HSPACE="20" VSPACE="20" ALIGN="RIGHT" BORDER="5"
ALT="Стандартная комплектация настольного персонального
компьютера">
```

Тогда, если навести указатель мыши на иллюстрацию, расположенную в верхней части страницы, и задержать указатель там на несколько секунд, то на данной иллюстрации появится следующая надпись: **«Стандартная комплектация настольного персонального компьютера»**. Используя этот атрибут, следует иметь в виду, что он распознается не всеми браузерами. Например, Internet Explorer «знает» этот атрибут, а Mozilla Firefox – нет.

3.3. Использование графики в качестве фона страницы

Остановимся еще на одной возможности использования графических элементов при оформлении Web-страниц. В главе 1 было изложено, как с помощью атрибута **BGCOLOR** тега **<BODY>** можно задавать фоновый цвет для страницы. Но фон, создаваемый с помощью этого атрибута, может иметь только один цвет.

Как же быть в том случае, если разработчик страницы хочет создать для нее разноцветный фон, использовать в качестве фона какой-либо узор? В этом случае следует воспользоваться другим атрибутом того же тега **BODY**. Это атрибут **BACKGROUND**. Алгоритм использования этого атрибута следующий:

а) за основу для создания фона берется определенный графический файл, содержащий некоторый рисунок, узор, орнамент. Если этот файл имеет формат **bmp** или другой формат, отличный от общепринятых в HTML, его следует заново сохранить в формате **jpg** или **gif**;

б) тег **<BODY>** дополнить атрибутом **BACKGROUND**. В качестве значения данного атрибута следует указать имя того файла, который содержит фоновый рисунок. Пример указания атрибута:

Вопросы для самоконтроля

1. Какие форматы рекомендуется использовать для графических файлов, которые должны стать элементами Web-страниц?
2. Какой вид тега (одиночный или парный) применяется для вставки нового графического элемента на Web-страницу?
3. Как преобразовать изображение к тому формату, который подходит для использования этого изображения в качестве фонового элемента Web-страницы?
4. Какой тег позволяет использовать графический файл для создания фона Web-страницы?
5. Какие способы взаимодействия текста и графики применяются на Web-странице?
6. Каким образом можно задать горизонтальный и вертикальный размеры изображения на Web-странице?
7. Каким образом можно задать альтернативный текст, выводимый в том случае, если иллюстрация не отображается при загрузке страницы?
8. Как создать рамку для выводимого изображения?
9. Как задать расстояние по вертикали и по горизонтали между изображением и текстом?
10. В каких единицах измеряются размеры окна для вставки изображения на странице?
11. Что происходит в том случае, если размеры окна, заданного для отображения иллюстрации, отличаются от фактических размеров этой иллюстрации?
12. В каких единицах измеряется расстояние по вертикали и по горизонтали между изображением и текстом?

ГЛАВА 4. Использование гиперссылок

В предыдущих главах пособия мы рассмотрели базовые элементы оформления Web-страниц, к которым относится форматирование текста на уровне символов и на уровне абзаца, использование для структурирования страницы заголовков нескольких уровней, разделительных линий и списков различного вида, вставка в документ элементов компьютерной графики.

Однако все эти элементы оформления применялись к отдельной, автономной Web-странице. На практике же в большинстве случаев Web-страница создается как часть определенного сайта, причем эта страница должна быть связана с другими страницами того же сайта. Кроме того, эта страница может иметь связи с Web-страницами, расположенными на других сайтах, и на других компьютерах, подключенных к глобальной сети. А физически эти компьютеры могут находиться в другом городе, другой стране или вообще на другом континенте.

Для обеспечения такого взаимодействия между компьютерами в языке HTML предусмотрен специальный механизм, представляющий систему **гиперссылок**. Гиперссылка представляет собой элемент Web-страницы, который содержит информацию об адресе другой страницы. В качестве такого элемента может выступать фрагмент текста или графический объект.

При наведении указателя мыши на гиперссылку указатель меняет свою обычную форму. Как правило, указатель принимает форму кисти руки с вытянутым пальцем. Щелчок по гиперссылке позволяет загрузить в браузере другую страницу, адрес которой содержится в этой ссылке.

4.1. Текстовые гиперссылки

Начнем изучение гиперссылок с текстового варианта. Гиперссылка создается с помощью парного тега **<A>** с атрибутом **HREF**. Общий вид тега следующий:

```
<A HREF="name.html">  
Текст гиперссылки  
</A>
```

Данный тег создает гиперссылку, щелчок по которой осуществляет переход на страницу с именем **name.html**. Между открывающей и закрывающей частью тега можно вводить любой текст, который и представляет собой собственно гиперссылку. Для того чтобы пользователь мог отличить гиперссылку от обычного текста, имеющегося на странице, эта ссылка автоматически выделяется цветом, отличным от цвета прочего текста, и подчеркиванием.

Приведенный вид гиперссылки представляет ее простейший вариант, когда страница, на которую происходит переход по ссылке, находится на компьютере в той же папке, что и сама исходная Web-страница.

Посредством гиперссылки можно выполнять переход и на страницы, расположенные в других папках компьютера. В этом случае, помимо имени самой страницы, нужно указать **путь** к этой странице.

Например, если страница, к которой требуется перейти по ссылке, находится в папке, вложенной в текущую (т.е. в папке нижнего уровня по отношению к текущей), то открывающий тег гиперссылки изменится следующим образом:

```
<A HREF="folder/name.html">
```

где **folder** – имя нижележащей папки, которое отделяется символом «слэш» от имени страницы.

Если же страница, указываемая в теге, расположена по отношению к текущей в вышележащей папке, то открывающая часть тега для перехода по ссылке будет выглядеть так:

```
<A HREF=" ../name.html">
```

Две горизонтальные точки в этом теге обозначают переход на один уровень вверх по файловой структуре компьютера.

В том случае, если Web-страница, на которую производится переход по гиперссылке, находится на другом компьютере, необходимо указывать полный путь к этой странице. Полный путь включает следующие элементы: название используемого протокола передачи данных, адрес сайта, на котором находится страница, и имя самой страницы. Общий вид соответствующего открывающего тега следующий:

```
<A HREF="http://www.site.com/page.html">
```

где **http** – название протокола передачи гипертекста, **www.site.com** – адрес сайта, **page.html** – имя страницы, на которую производится переход. Если имя страницы в теге не указано, то по гиперссылке происходит переход на главную страницу сайта. Как правило, эта страница носит имя **index.html**.

Рассмотрим использование гиперссылок на следующем примере: требуется создать географический справочник, в котором содержится информация о ряде крупнейших стран мира. Данный справочник представляет собой сайт, состоящий из одной главной страницы и нескольких страниц, каждая из которых посвящена одной из стран.

Главная страница содержит общую информацию о содержании и назначении справочника, гиперссылки на страницы, посвященные отдельным странам, и ссылки на самые популярные во всемирной сети поисковые системы, необходимые в том случае, если пользователю понадобится дополнительная информация.

Каждая страница сайта с информацией по отдельной стране содержит такие данные, как географическое положение страны, численность населения, форма государственного устройства, государственный язык,

валюта, столица государства, список крупнейших городов. Кроме того, для удобства навигации (перемещения по страницам сайта) на каждой такой странице должна быть ссылка, позволяющая вернуться на главную страницу сайта.

Разберем более подробно структуру главной страницы сайта. Эта страница имеет традиционное для главной страницы сайта имя **index.html**. В верхней части страницы находится заголовок первого уровня с названием сайта. Под заголовком расположена иллюстрация, представляющая собой изображение географического глобуса, выровненное по левому краю страницы. Справа от изображения находится текст, содержащий общую информацию о тематике и содержании сайта.

Ниже находятся гиперссылки, ведущие на 4 страницы, каждая из которых содержит информацию об одной из стран. Для того, чтобы содержание страницы было лучше структурировано каждая гиперссылка помещается в отдельный абзац. Первая ссылка ведет на страницу **P_Russia.html**, посвященную России, вторая – на страницу **P_USA.html** с информацией о США, третья – на страницу **P_France.html** о Франции, четвертая – на страницу **P_Japan.html** о Японии.

Под гиперссылками, ведущими на страницы сайта, помещаются еще 2 гиперссылки на популярные поисковые системы **Яндекс** и **Google**. Ниже приводится HTML-код главной страницы сайта.

```
<HTML>
<HEAD>
<TITLE>
Работа с гиперссылками
</TITLE>
</HEAD>
<BODY BGCOLOR="CCFFCC" TEXT="BLUE">
<DIV ALIGN="CENTER">
<H1>
Географический справочник
</H1>
</DIV>
<DIV ALIGN="JUSTIFY">
<IMG SRC="travel.jpg" ALIGN="LEFT" height="170"
width="180" VSPACE="5" HSPACE="5" BORDER="4">
<FONT SIZE="+1">
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
```

Данный сайт содержит информацию по нескольким крупнейшим странам мира. Для каждой страны представлены такие сведения, как географическое положение, форма государственного устройства, площадь, численность населения, государственный язык, валюта, крупнейшие города.

перед текстом, занятое пробелами, также будет подчеркнуто, что испортит внешний вид страницы.

Внешний вид главной страницы сайта – географического справочника приведен на рис.4.1.

По умолчанию текст, неиспользованной ранее гиперссылки, как правило, выделяется синим цветом, текст активной ссылки – красным, а уже посещенной ранее ссылки – лиловым. Но пользователь имеет возможность изменить эти цвета с помощью дополнительных атрибутов тега **<BODY>**. Для этой цели нужно использовать атрибуты **LINK**, **ALINK** и **VLINK**. Атрибут **LINK** определяет основной цвет ссылки, **ALINK** – активной ссылки, **VLINK** – ранее посещенной ссылки. Значением каждого из этих атрибутов может быть заключенное в кавычки название цвета или его шестнадцатеричный код.

Если на нашей странице изменить тег **<BODY>** следующим образом:

```
<BODY BGCOLOR="CCFFCC" TEXT="BLUE"  
LINK="GREEN" ALINK="GOLD" VLINK="MAROON">
```

то неиспользованные ссылки будут выделены зеленым цветом, активные – золотистым, а использованные – бордовым.

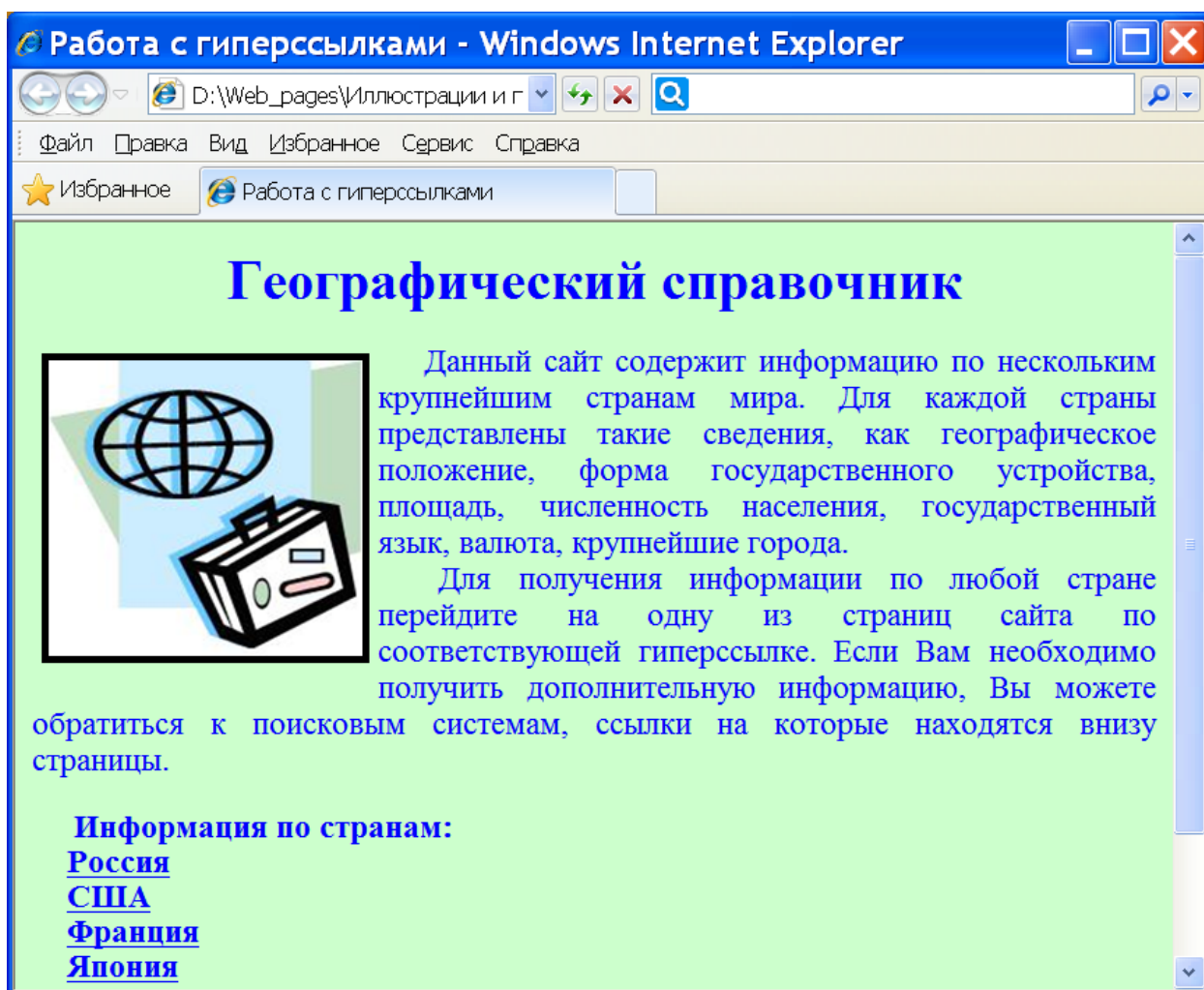


Рис. 4.1. Главная страница сайта «Географический справочник»

Теперь создадим 4 страницы для каждой из описанных в справочнике стран. Подробно на структуре данных страниц останавливаться не будем, так как она уже была описана выше. Отметим только следующее обстоятельство: хотя, как правило, на панели инструментов браузера имеются кнопки для возвращения на предыдущую страницу, лучше создать подобную гиперссылку и на самой Web-странице. Это делается для обеспечения удобства пользователя, просматривающего страницу. Изучив все содержимое страницы, пользователь внизу этой страницы сразу видит ссылку для возврата на главную страницу.

В качестве примера оформления ниже приводится текст страницы сайта **P_Russia.html**, посвященной России.

```
<HTML>
<HEAD>
<TITLE>
Справочник. Страница 1
</TITLE>
</HEAD>
<BODY BGCOLOR="CCFFCC" TEXT="BLUE"
LINK="GREEN" ALINK="GOLD" VLINK="MAROON">
<DIV ALIGN="CENTER">
<H1>
<FONT COLOR="RED">
<IMG SRC="Flag_RF.jpg" WIDTH="99" HEIGHT="66">
Россия
</FONT>
</H1>
</DIV>
<HR SIZE="5" COLOR="MAROON">
<DIV>
<FONT SIZE="+1">
<STRONG>
Российская федерация - государство, расположенное в
Восточной Европе и Северной Азии.
</STRONG>
</FONT>
</DIV>
<DIV>
<FONT SIZE="+1">
<STRONG>
Население - 146 млн чел.
</STRONG>
</FONT>
</DIV>
<DIV>
```

```

<FONT SIZE="+1">
<STRONG>
Территория - 17 млн км<SUP>2</SUP>
</STRONG>
</FONT>
</DIV>
<DIV>
<FONT SIZE="+1">
<STRONG>
Государственное устройство - федеративная
республика, включающая в свой состав 85 субъектов.
</STRONG>
</FONT>
</DIV>
<DIV>
<FONT SIZE="+1">
<STRONG>
Государственный язык - русский.
</STRONG>
</FONT>
</DIV>
<DIV>
<FONT SIZE="+1">
<STRONG>
Валюта - рубль.
<IMG SRC="rouble.jpg">
</STRONG>
</FONT>
</DIV>
<DIV>
<FONT SIZE="+1">
<STRONG>
Столица - Москва (население - 12,2 млн чел.)
</STRONG> </FONT> </DIV>
<DIV>
<FONT SIZE="+1">
<STRONG>
Крупнейшие города:
<UL>
<LI> Санкт-Петербург (5,2 млн чел.)
<LI> Новосибирск (1,6 млн чел.)
<LI> Екатеринбург (1,4 млн чел.)
<LI> Нижний Новгород (1,3 млн чел.)
</UL>
</STRONG>

```

```

</FONT>
</DIV>
<DIV>
<A HREF="Index.html">
<FONT SIZE="+1">
<STRONG>
На главную страницу
</STRONG>
</FONT>
</A>
</DIV>
</BODY>
</HTML>

```

На рис. 4.2 показан внешний вид страницы сайта, посвященной Российской Федерации.

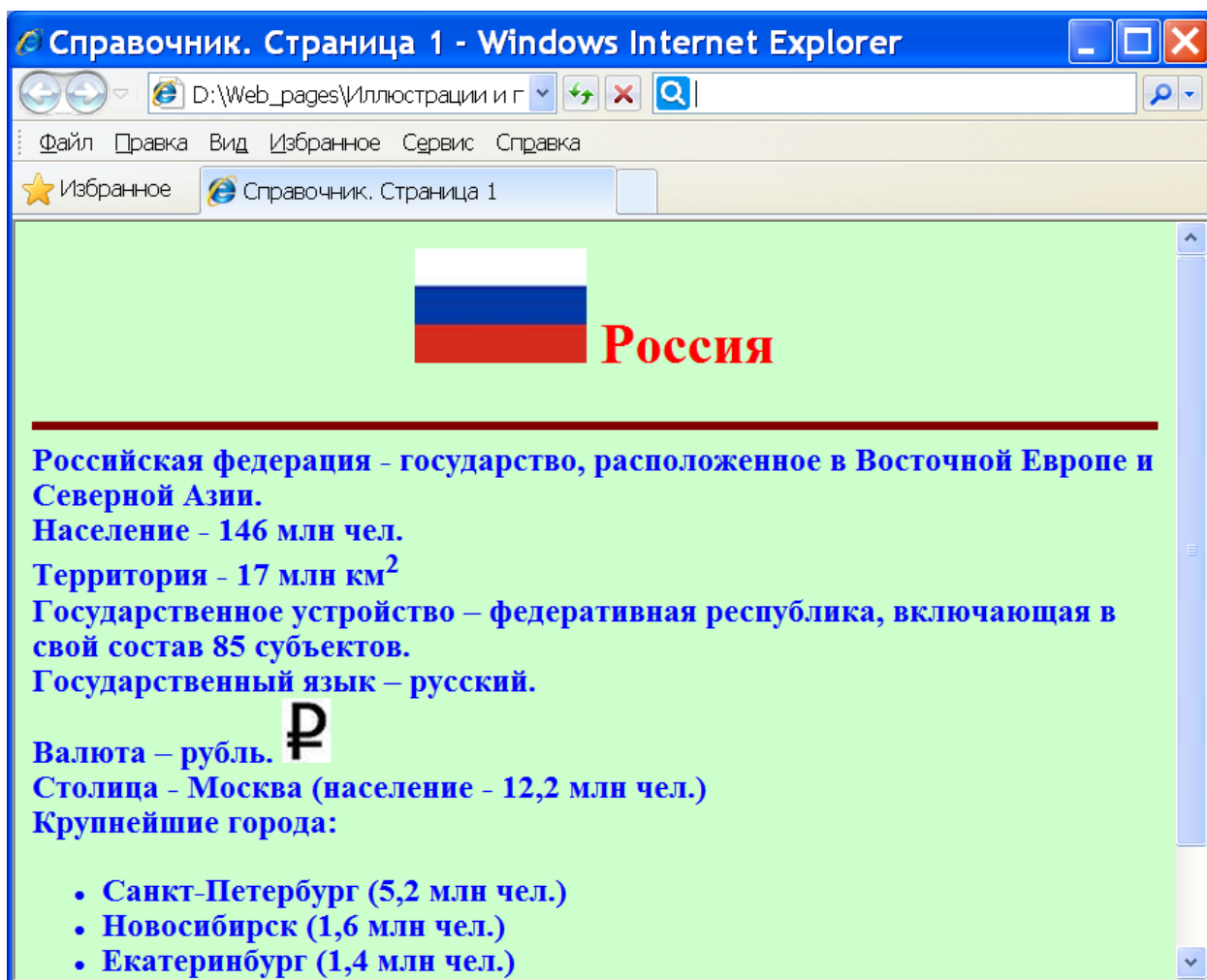


Рис. 4.2. Страница сайта «Географический справочник», посвященного Российской Федерации

4.2. Графические гиперссылки

В начале этой главы уже упоминалось о том, что в качестве гиперссылки может использоваться не только текст, но и элемент компьютерной графики, т.е. какая-либо иллюстрация, рисунок, картинка.

Для создания графической гиперссылки используется тот же парный тег **<A>** с атрибутом **HREF**, что и для создания текстовой. Отличие заключается в том, что между открывающей и закрывающей частями тега **<A>** вставляется не текст, а другой, одиночный тег ****. Как известно, тег **** используется для того, чтобы загрузить на страницу какое-либо изображение. Это изображение и становится ссылкой, с помощью которой выполняется переход с текущей страницы сайта на другую страницу. Таким образом, общий вид тега, создающего графическую ссылку следующий:

```
<A HREF="page.html">
<IMG SRC="picture.jpg">
</A>
```

Теперь рассмотрим создание графической ссылки на практике. Для этого вернемся на главную страницу сайта «Географический справочник». В папку, содержащую страницы сайта, помещаем еще 4 графических файла, взятых из коллекции клипартов **Microsoft Office**.

Файл с изображением дома в лесу на берегу озера назовем **forest.jpg**. Это файл используем для ссылки на страницу с информацией о России. Для создания ссылки на страницу о США используем файл с изображением небоскребов, который назовем **skyscraper.jpg**. Для создания ссылки на страницу о Франции используем иллюстрацию с Эйфелевой башней, которой дадим имя **tower.jpg**. Переход на страницу о Японии будет выполняться по картинке с изображением пагоды, которую получит имя **pagoda.jpg**.

Вставляем в тег **<A>** с атрибутом **HREF**, создающий ссылку на страницу о России, тег ****, загружающий файл **forest.jpg**. Тем самым мы к текстовой ссылке добавляем ссылку в графической форме, и тег изменяется следующим образом:

```
<A HREF="P_RUSSIA.HTML">
<IMG SRC="forest.jpg" WIDTH="120" HEIGHT="100">
<FONT SIZE="+1"> <STRONG>
Россия
</STRONG> </FONT>
</A>
```

В результате на главной странице сайта рядом появляются две ссылки на страницу о России: одна в текстовой форме, а другая в графической.

Затем аналогичным образом изменяем тег со ссылкой на страницу о США, помещая в него вложенный тег **IMG SRC** со ссылкой на файл **skyscraper.jpg**. Внутри тега для перехода на страницу о Франции вставляем ссылку на файл **tower.jpg** и в тег для перехода на страницу о Японии вставляем ссылку на **pagoda.jpg**.

В результате внесенных в код главной страницы изменений она приобретает вид, показанный на рис. 4.3.

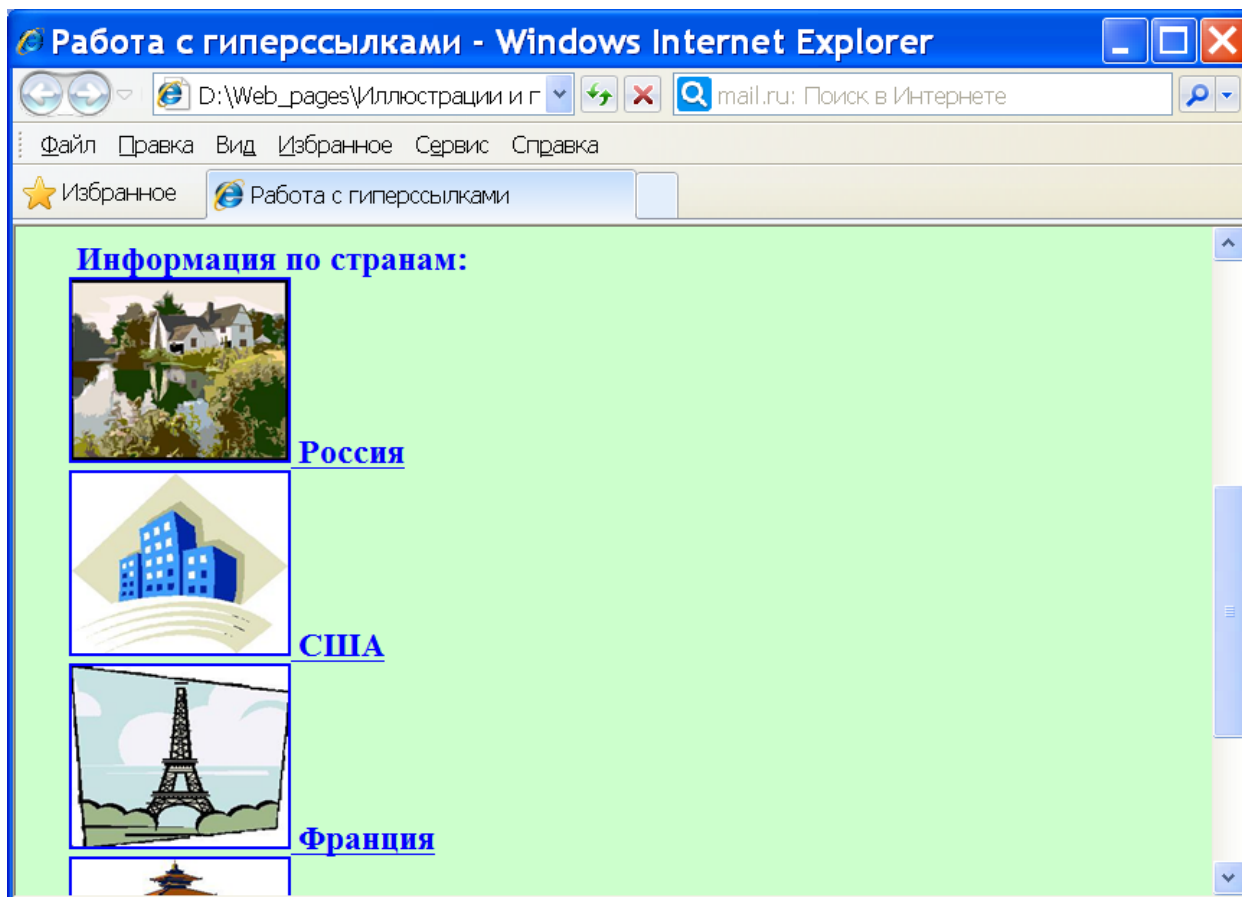


Рис. 4.3. Главная страница сайта «Географический справочник» после добавления в нее графических ссылок

Рис. 4.3 наглядно демонстрирует, что теперь каждая текстовая ссылка, ведущая на одну из страниц сайта, продублирована соответствующей графической ссылкой. После внесения изменений в HTML-код страницы следует еще раз пройти по всем имеющимся на сайте ссылкам, чтобы убедиться, что они корректно работают.

4.3. Закладки и внутристраничные гиперссылки

Гиперссылки в языке HTML представляют собой универсальное средство, который может успешно применяться не только для перемещения между страницами, но и для навигации внутри самой страницы. Этот вопрос становится особенно актуальным, если Web-страница имеет большой объем и ее содержимое занимает несколько экранов. В такой длинной странице

пользователю иногда бывает трудно ориентироваться, и в этом случае на помощь ему приходит система внутривстраничных ссылок.

Механизм создания и использования внутривстраничных ссылок заключается в следующем: внутри страницы создаются якоря (закладки), на которые можно производить переход по ссылке. Для создания якоря используется парный тег, имеющий следующий общий вид:

```
<A NAME="anchor">  
Текст закладки  
</A>
```

Таким образом, для создания закладки используется уже знакомый нам тег **<A>**, но с другим атрибутом. Вместо атрибута **HREF** используется атрибут **NAME**, значением которого является имя создаваемой закладки. Это имя может содержать только латинские буквы и цифры. Внутри тега находится собственно текст якоря (закладки). Этот текст может содержать любые символы, которые имеются на клавиатуре компьютера, т.е. буквы кириллицы, латинские буквы, цифры, знаки препинания, различные специальные символы.

К этой закладке можно перейти из любого другого места Web-страницы, для чего в этом месте нужно создать **ссылку на якорь** (закладку). Ссылка на закладку внутри страницы похожа на ссылку на внешнюю страницу, но имеет одно отличие. Общий вид такой ссылки следующий:

```
<A HREF="#anchor">  
Текст ссылки  
</A>
```

Гиперссылка на якорь использует тот же тег **<A>** с тем же атрибутом **HREF**, что и ссылка на внешнюю страницу. Но в данном случае гиперссылка производится на закладку внутри страницы. Для того чтобы можно было отличить внешнюю и внутреннюю гиперссылки друг от друга, в языке HTML принято следующее соглашение: в ссылке на имя внутривстраничной закладки перед ее именем, заключенным в кавычки, ставится символ # (октоторп). Октоторп показывает, что перемещение по ссылке происходит внутри текущей страницы.

Рассмотрим использование закладок и внутривстраничных ссылок на следующем примере. Необходимо создать Web-страницу, в которой рассказывается о наиболее широко используемых языках программирования. По каждому языку дается информация справочного характера. Сообщается о времени создания языка, приводятся имена авторов языка, рассказывается о сфере применения языка и некоторых других его особенностях. Поскольку информация, имеющаяся на данной странице, не помещается на одном экране, то для удобства навигации по странице сделано следующее: в начале страницы имеется список языков программирования. Каждый элемент списка – название одного из языков. Это название представляет собой гиперссылку

на раздел страницы с описанием данного языка. Например, ссылка на раздел страницы, посвященный языку Паскаль, выглядит следующим образом:

```
<LI> <A HREF="#Pascal">Паскаль </A>
```

В данном случае одиночный тег **LI** говорит о том, что эта ссылка является элементом списка, **Pascal** – название закладки, на которую происходит переход по ссылке, **Паскаль** – текст ссылки, который увидит на экране пользователь при просмотре страницы в браузере.

Раздел страницы, посвященный языку Паскаль, начинается с заголовка второго уровня, который и является в данном случае закладкой. Вид этой закладки в HTML-коде следующий:

```
<H2> <FONT COLOR="GREEN">  
<A NAME="Pascal"> Паскаль </A>  
</FONT> </H2>
```

Тег **H2** показывает, что текст является заголовком второго уровня. Тег **FONT** выделяет заголовок зеленым цветом (в отличие от синего, которым написан основной текст страницы). Тег **A** с атрибутом **NAME** создает собственно саму закладку, причем **Pascal** – это имя закладки, а **Паскаль** – текст закладки, который пользователь увидит в браузере.

Для того чтобы пользователь, изучив справочный материал по Паскалю, мог затем легко найти информации по любому другому языку из числа представленных на странице, сделано следующее. В конце раздела по языку Паскаль имеется ссылка на начало страницы. По этой ссылке пользователь переходит на закладку с именем **Begin**. Эта закладка представляет собой заголовок первого уровня, выделенный красным цветом и находящийся непосредственно над списком языков. Перейдя из раздела о языке Паскаль по вышеуказанной ссылке, пользователь вновь видит перед собой список языков программирования и может теперь перейти в раздел с информацией о любом другом языке (Бейсике, Си, Фортране), выбрав из списка соответствующую ссылку. Полный HTML-код страницы о языках программирования приведен ниже:

```
<HTML>  
<HEAD>  
<TITLE>  
Создание ссылок внутри документа  
</TITLE>  
</HEAD>  
<BODY BGCOLOR="CCFFCC" TEXT="BLUE">  
<DIV ALIGN="CENTER">  
<H1> <FONT COLOR="RED">  
<A NAME="Begin"> Языки программирования </A>  
</FONT> </H1>
```


программного обеспечения для решения широкого круга задач. Синтаксис языка Си в дальнейшем стал основой для таких популярных языков программирования, как C++, C#, Java

</DIV>

<DIV ALIGN="CENTER">

 В начало страницы

 </DIV>

</BODY>

</HTML>

На рис.4.4 показано начало страницы, содержащей справочную информацию о языках программирования, со списком языков.

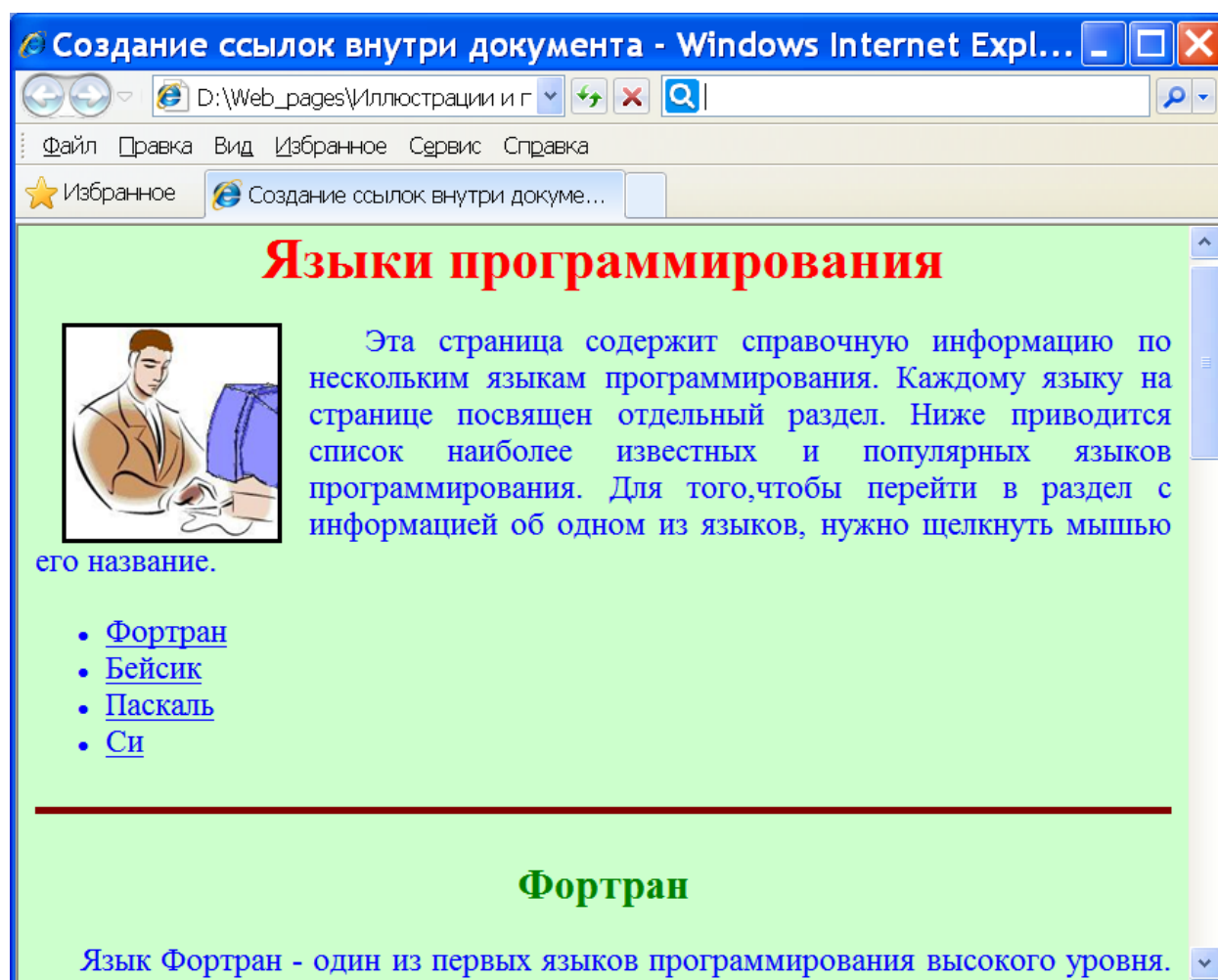


Рис. 4.4. Начало страницы, посвященной языкам программирования, со списком языков, информация о которых присутствует на странице

Из приведенной выше иллюстрации (рис.4.4) понятно, что, открыв страницу, пользователь видит только список языков и начало раздела, в котором описывается язык Фортран. Конечно, пользователь может ознакомиться со всем текстом страницы с помощью вертикальной полосы

прокрутки, но есть более быстрый и удобный способ. Для быстрого получения информации по другим языкам следует воспользоваться списком ссылок. В этом списке можно выбрать интересующий пользователя язык и попасть в нужный раздел одним щелчком мыши.

На рис. 4.5. представлен раздел страницы о языках Паскаль и Си.

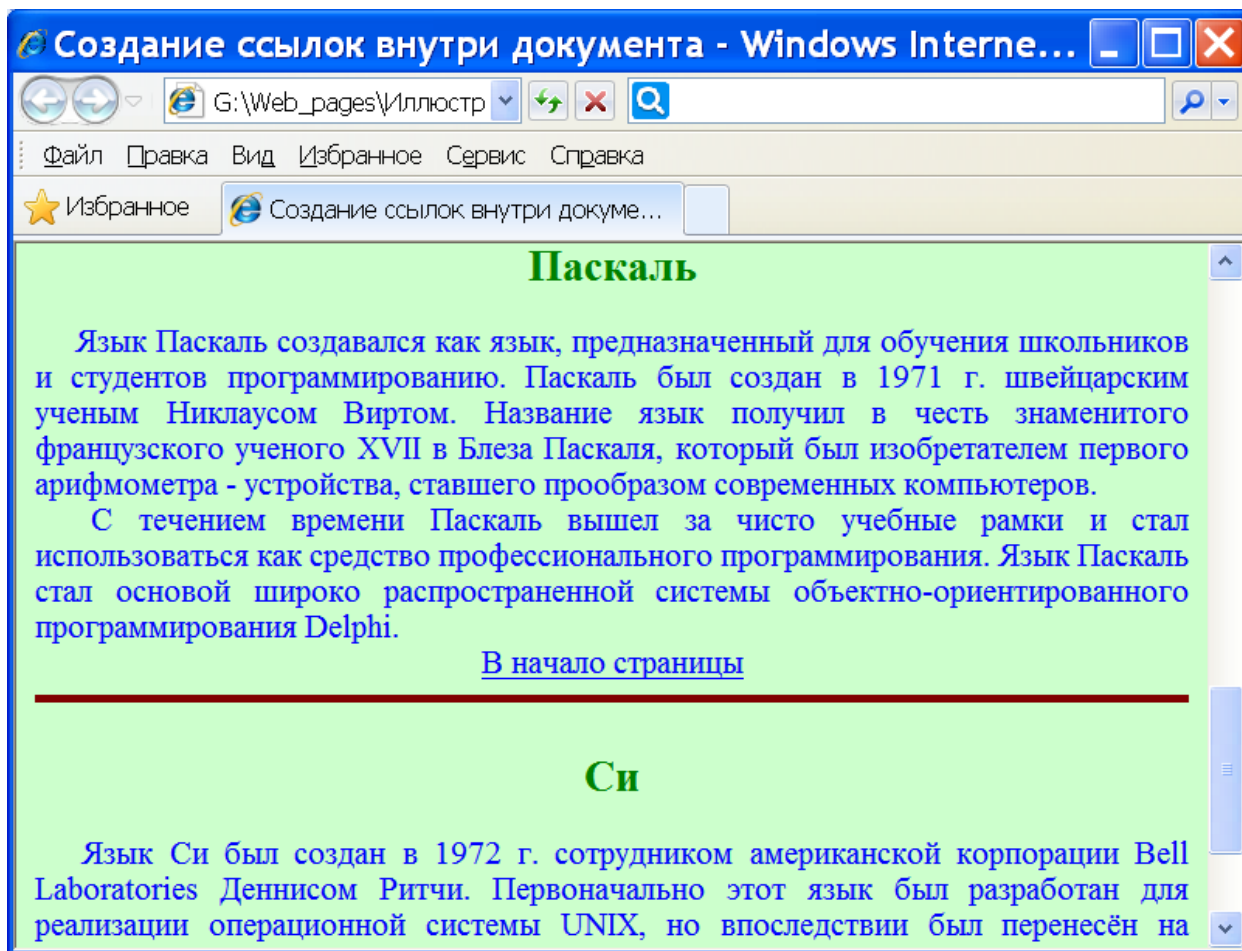


Рис. 4.5. Раздел страницы, посвященной, языкам программирования, содержащий информацию о языках Паскаль и Си

Изучив разделы, посвященные Паскалю и Си, пользователь может прибегнуть к помощи гиперссылки с текстом «**В начало страницы**». Один раз щелкнув мышью, пользователь вновь оказывается вверху страницы и там может выбрать из списка другой нужный ему раздел.

Следует также отметить, что внутренние гиперссылки можно использовать не только внутри страницы. В том случае, если создавая внешнюю гиперссылку на страницу, мы хотим, чтобы пользователь попадал не в начало этой страницы, а сразу в определенный раздел, мы к внешней гиперссылке добавляем название закладки. Пример подобной ссылки:

```
<A HREF="Languages.html#Pascal">
```

Переход на страницу о языках программирования в раздел о языке Паскаль

```
</A>
```

Обратите внимание, что имя страницы от имени закладки отделяется тем же октоторпом, что и для внутренней ссылки. Переход по такой ссылке с какой-либо внешней страницы на созданную нами страницу о языках программирования позволит сразу же, не используя список, попасть в нужный раздел этой страницы (в данном случае – это раздел о языке Паскаль).

Вопросы для самоконтроля

1. Как отличить по внешнему виду гиперссылку от остального текста Web-страницы?
2. Какой атрибут изменяет цвет неиспользованной ссылки?
3. Какой атрибут изменяет цвет активной ссылки?
4. Какой атрибут изменяет цвет посещенной ранее ссылки?
5. Какой тег создает текстовую гиперссылку?
6. Какой тег создает графическую гиперссылку?
7. Является ли тег для создания гиперссылки одиночным или парным?
8. В каком случае в гиперссылке можно указать только имя страницы, на которую производится переход, не указывая полный путь к ней?
9. В каком случае при создании гиперссылки нужно указать полный путь к странице, на которую производится переход по этой ссылке?
10. Какое имя, как правило, используется для главной страницы сайта?
11. Для чего в тексте Web-страницы создаются закладки (якоря)?
12. Какой тег используется для того, чтобы создать ссылку на внутристраничную закладку?

ГЛАВА 5. Работа с таблицами

Одним из основных вопросов, которой необходимо решать разработчику Web-страницы – это регулировка взаимного расположения различных элементов текста и графики на странице. Частично эта проблема может быть решена с помощью дополнительных атрибутов тега ****, о которых было изложено в главе 3 пособия.

Но существует и другой, более универсальный способ решения этой проблемы. Это использование таблиц, которые в языке HTML могут использоваться не только как средство наглядного представления определенных численных данных, но и как способ задать нужную внутреннюю структуру самой страницы.

Внешний вид таблицы может значительно отличаться в зависимости от конкретной цели, для которой она может применяться: таблица может иметь или не иметь видимую рамку, у таблицы может присутствовать или отсутствовать заголовок, в таблице может быть или не быть фон определенного цвета. Фон таблицы, в свою очередь, может быть одноцветным, или в качестве фона может быть использован какой-либо рисунок. Но вне зависимости от особенностей внешнего вида таблицы она может успешно выполнять свое основную функцию по упорядочению всей информации, представленной на Web-странице.

5.1. Создание простой таблицы

Таблица в HTML создается с помощью парного тега **<TABLE>...</TABLE>**. Как и многие другие теги, **<TABLE>** обладает рядом дополнительных атрибутов. Основными атрибутами тега являются:

- а) **BORDER** – ширина рамки, выраженная в пикселях;
- б) **BGCOLOR** – фоновый цвет таблицы;
- в) **HEIGHT** и **WIDTH** – высота и ширина таблицы.

В состав таблицы входят следующие основные элементы, каждый из которых создается особым парным тегом:

I. Заголовок таблицы, который создается с помощью тега **<CAPTION>...</CAPTION>**. Этот тег должен быть вложен в тег **<TABLE>**. Тег **<CAPTION>**, в отличие от других элементов, является необязательным, т.е. заголовок у таблицы может отсутствовать, если в нем нет необходимости. Данный тег имеет дополнительный атрибут **ALIGN**, с помощью которого определяется местоположение заголовка относительно самой таблицы. Этот атрибут имеет всего два значения: значение **TOP** определяет положение заголовка над таблицей (этот способ расположения заголовка применяется по умолчанию); значение **BOTTOM** задает расположение заголовка под таблицей.

II. Строка, создаваемая с помощью тега **<TR>...</TR>**. Этот тег, подобно тегу **<CAPTION>**, должен быть вложен в тег **<TABLE>**. Данный тег имеет следующие основные атрибуты:

а) **ALIGN** – выравнивание текста по горизонтали. В отличие от выравнивания текста в абзаце, в строке данный атрибут может принимать только три значения: **LEFT** – выравнивание текста по левому краю, **CENTER** – по центру, **RIGHT** – по правому краю, т.е. выравнивание по ширине, применяемое в абзаце, в таблице невозможно;

б) **VALIGN** – выравнивание текста по вертикали. Этот атрибут может также принимать три значения: **TOP** – выравнивание по верхнему краю строки, **BOTTOM** – по низу строки, **MIDDLE** – по середине;

в) **BGCOLOR** – фоновый цвет строки.

III. Ячейка, создаваемая с помощью тега **<TD>...</TD>**. Так как ячейки являются составной частью строки, то тег **<TD>** должен быть вложен в тег **<TR>**. Для данного тега основными являются те же атрибуты **ALIGN**, **VALIGN** и **BGCOLOR**, только область действия этих тегов распространяются не на всю строку, а только на ту ячейку, в которой они используются.

Таким образом, для создания таблицы, содержащей заголовок, расположенный над основной частью таблицы, 2 строки и по 3 ячейки в каждой строке, используется тег следующего вида:

```
<TABLE>
<CAPTION> ... <CAPTION>
<TR>
<TD>...</TD>  <TD>...</TD>  <TD>...</TD>
</TR>
<TR>
<TD>...</TD>  <TD>...</TD>  <TD>...</TD>
</TR>
</TABLE>
```

Многоточиями обозначено содержимое ячеек. Таким содержимым может быть текст и изображения, причем и то и другое может использоваться в качестве гиперссылок.

Рассмотрим создание простой таблицы средствами HTML на следующем примере. На странице, посвященной программному пакету Microsoft Office, необходимо создать таблицу, в которой были бы указаны названия основных программ, входящих в данный пакет, и их характеристики.

Таблица будет включать в себя заголовок и 2 строки. В первой строке будут перечислены названия программ, а во второй – назначение каждой программы. В каждой строке должна быть одна ячейка, в которой содержится заголовок строки, и 4 ячейки, каждая из которых соответствует одной из перечисляемых в таблице программ. Таким образом, строки

таблицы будут содержать по 5 ячеек. Ниже приводится HTML-код для создания Web-страницы с такой простой таблицей:

[illegible]

Данная страница посвящена программному пакету Microsoft Office. Этот пакет был разработан специалистами американской фирмы Microsoft для обработки различных типов документов. Пакет также включает встроенную систему программирования VBA.

[illegible]

В приведенной ниже таблице перечислены основные программы, входящие в состав пакета, и кратко описано их назначение.

```
</FONT>
</DIV>
<BR>
<TABLE WIDTH="90%">
<CAPTION>
<FONT SIZE="5">
<STRONG>
Основные программы пакета Microsoft Office
</STRONG>
</FONT>
</CAPTION>
<TR>
<TD>
<STRONG> <FONT SIZE="4">
```

```

Название программы
</FONT> </STRONG>
</TD>
<TD> <FONT SIZE="4">
MS Word
</FONT> </TD>
<TD> <FONT SIZE="4">
MS Excel
</FONT> </TD>
<TD> <FONT SIZE="4">
MS Access
</FONT> </TD>
<TD> <FONT SIZE="4">
MS PowerPoint
</FONT> </TD>
</TR>
<TR>
<TD>
<STRONG> <FONT SIZE="4">
Характеристика программы
</FONT> </STRONG>
</TD>
<TD> <FONT SIZE="4">
Текстовый процессор
</FONT> </TD>
<TD> <FONT SIZE="4">
Табличный процессор
</FONT> </TD>
<TD> <FONT SIZE="4">
Система управления базами данных
</FONT> </TD>
<TD> <FONT SIZE="4">
Программа для работы с электронными презентациями
</FONT> </TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

На рис. 5.1 представлен внешний вид Web-страницы с таблицей. Основная задача, для которой была создана таблица, выполнена: информация об основных приложениях, входящих в состав пакета Microsoft Office, представлена в наглядном и упорядоченном виде. Но, в то же время, отсутствуют некоторые элементы оформления, которые могли бы сделать таблицу более выразительной, нет границы вокруг таблицы в целом и

отдельных ее ячеек, фон таблицы не отличается от основного фона Web-страницы, выравнивание текста во всех ячейках выполнено только по левому краю.

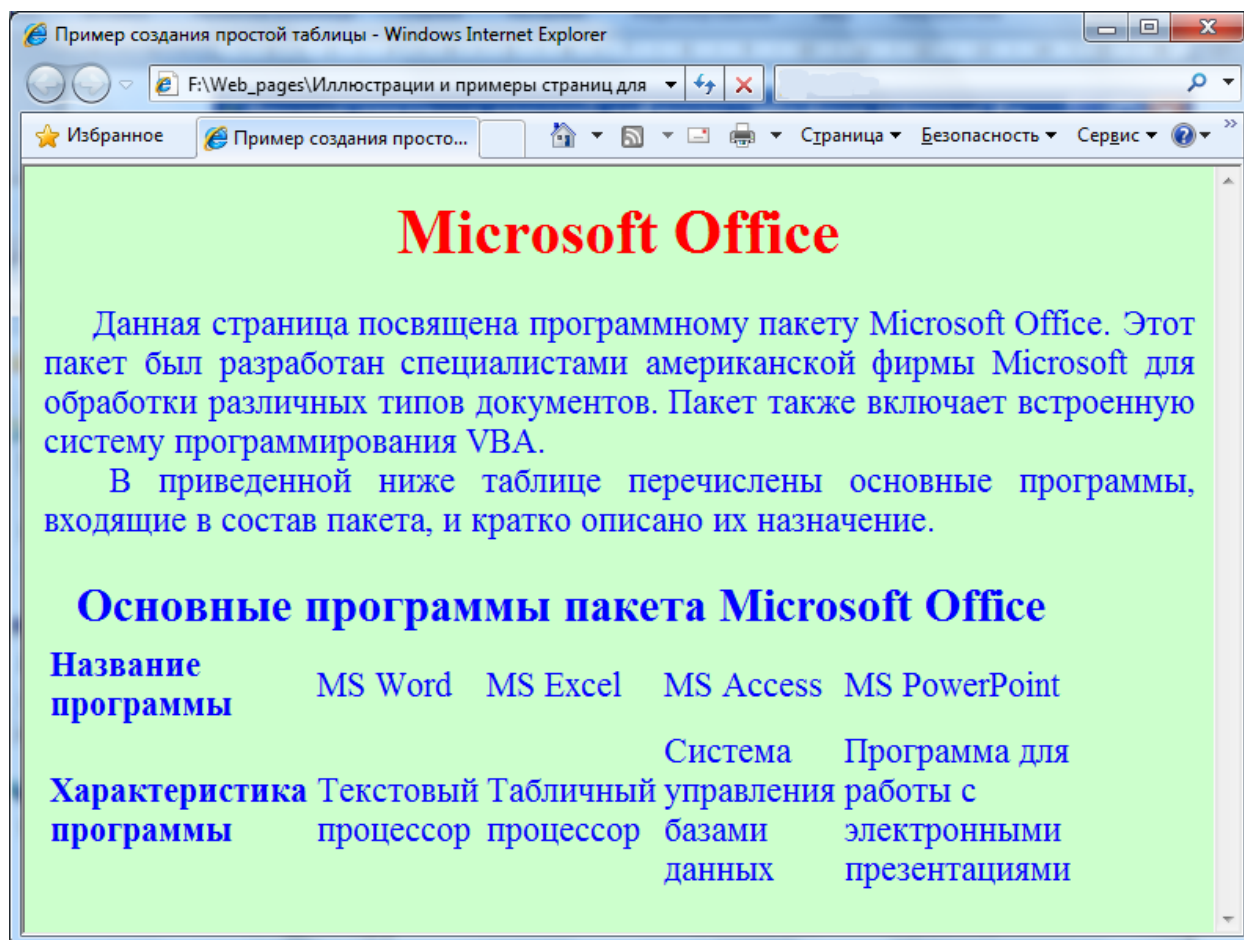


Рис.5.1. Страница, содержащая простую таблицу без оформления

Для того чтобы улучшить оформление таблицы, следует внести определенные изменения в HTML-код страницы. Для создания рамки добавляем в тег **<TABLE>** атрибут **BORDER** со значением, равным 3, т.е. толщина рамки составит 3 пикселя.

Кроме того, желательно создать фоновый цвет для каждой ячейки, имеющейся в таблице. Ячейки, содержащие заголовки строк, будут выделенные аквамариновым цветом. Ячейки, относящиеся к содержанию этих строк, будут выделены в шахматном порядке золотистым и розовым цветом. Для обеспечения нужного фона для каждой ячейки в открывающую часть тега **<TD>** добавляем атрибут **BGCOLOR**. Значением атрибута является соответствующий цвет, представленный в текстовой форме.

Еще одним элементом оформления, использованным в данной таблице, является способ выравнивания текста. По умолчанию текст в таблице выравнивается по левому краю. Мы же применим выравнивание по центру. Поскольку такой способ выравнивания использован во всей таблице, нет необходимости прописывать его отдельно в каждой ячейке. Вместо этого

достаточно для обеих строк таблицы в открывающую часть тега **<TR>** добавить атрибут **ALIGN** со значением **CENTER**.

Ниже приводится модифицированный фрагмент HTML-кода Web-страницы, относящийся к таблице (приводить целиком весь код нет необходимости, так как изменения внесены только в этот фрагмент, описывающий таблицу).

```
<TABLE WIDTH="90%" BORDER="3">
<CAPTION>
<FONT SIZE="5">
<STRONG>
Основные программы пакета Microsoft Office
</STRONG>
</FONT>
</CAPTION>
<TR ALIGN="CENTER">
<TD BGCOLOR="AQUA">
<STRONG> <FONT SIZE="4">
Название программы
</FONT> </STRONG>
</TD>
<TD BGCOLOR="GOLD"> <FONT SIZE="4">
MS Word
</FONT> </TD>
<TD BGCOLOR="PINK"> <FONT SIZE="4">
MS Excel
</FONT> </TD>
<TD BGCOLOR="GOLD"> <FONT SIZE="4">
MS Access
</FONT> </TD>
<TD BGCOLOR="PINK"> <FONT SIZE="4">
MS PowerPoint
</FONT> </TD>
</TR>
<TR ALIGN="CENTER">
<TD BGCOLOR="AQUA">
<STRONG> <FONT SIZE="4">
Характеристика программы
</FONT> </STRONG>
</TD>
<TD BGCOLOR="PINK"> <FONT SIZE="4">
Текстовый процессор
</FONT> </TD>
<TD BGCOLOR="GOLD"> <FONT SIZE="4">
Табличный процессор
```

```

</FONT> </TD>
<TD BGCOLOR="PINK"> <FONT SIZE="4">
Система управления базами данных
</FONT> </TD>

<TD BGCOLOR="GOLD"> <FONT SIZE="4">
Программа для работы с электронными презентациями
</FONT> </TD>
</TR>
</TABLE>

```

На рис. 5.2 показан внешний вид Web-страницы, посвященной программному пакету Microsoft Office, с внесенными в нее изменениями, касающимися оформления таблицы.

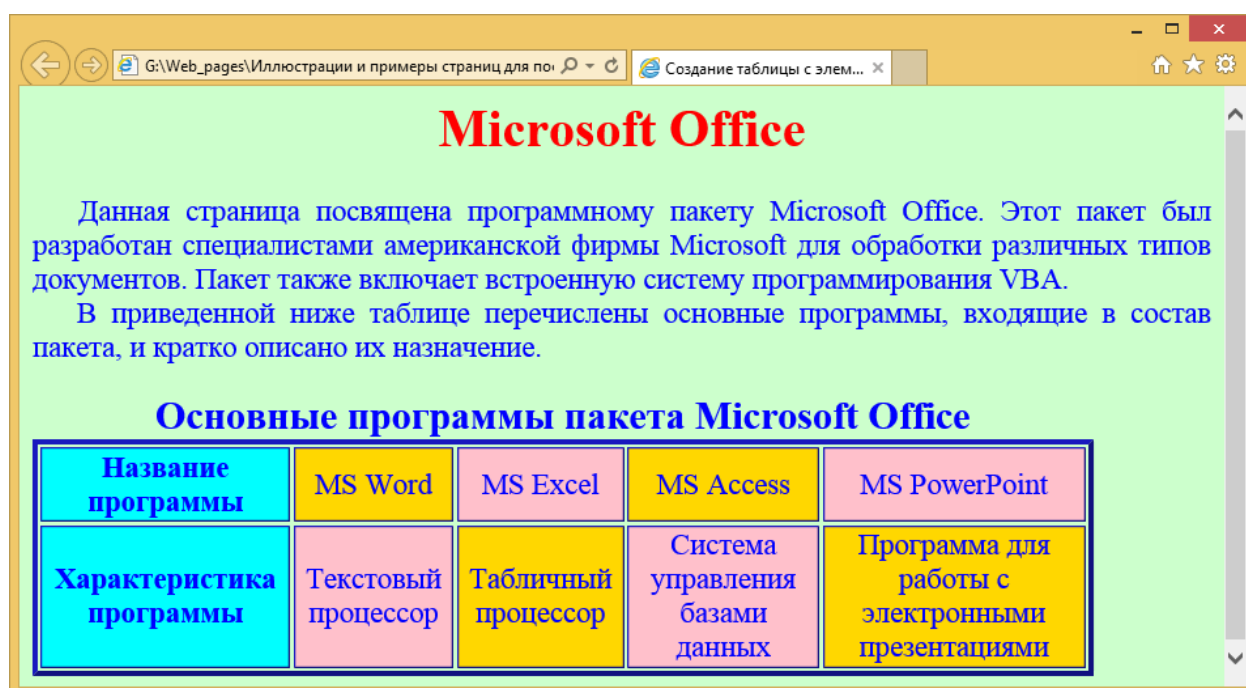


Рис. 5.2. Страница, содержащая таблицу с элементами оформления (рамка, фоновые цвета для ячеек, выравнивание текста)

При создании и оформлении таблиц нередко возникает необходимость объединения двух или более ячеек в одну. Язык HTML предоставляет такую возможность с помощью атрибутов, добавляемых в открывающую часть тега **<TD>** **COLSPAN** и **ROWSPAN**. Атрибут **COLSPAN** применяется для объединения ячеек по горизонтали, а атрибут **ROWSPAN** – для объединения по вертикали. Значениями обоих атрибутов является количество объединяемых ячеек, выраженное целым числом.

Например, если в теге **<TD>** используются следующие дополнительные атрибуты:

```
<TD COLSPAN="3">
```

то с помощью данного тега будут объединены 3 ячейки, расположенные по горизонтали.

Если же в теге **<TD>** будет использован другой дополнительный атрибут

<TD ROWSPAN="3">

то текущая ячейка «растянется» еще на 2 ячейки, которые распложены под ней и находятся в 2 нижележащих строках.

Рассмотрим применение этих дополнительных атрибутов на следующем примере. Необходимо создать Web-страницу, на которой рассматриваются основные классы прикладного программного обеспечения. К каждому из этих классов относятся несколько популярных приложений. Для наглядного отображения данной классификации на странице необходимо создать таблицу, содержащую 2 строки. Первая строка содержит названия классов программного обеспечения, а во второй строке перечислены конкретные виды этого обеспечения. Для создания такой страницы использован следующий HTML-код:

<HTML>

<HEAD>

<TITLE>

Объединение ячеек в таблице

</TITLE>

</HEAD>

<BODY BGCOLOR="#CCFFCC" TEXT="BLUE">

<DIV ALIGN="CENTER">

**<H1> **

Прикладное программное обеспечение

** </H1>**

</DIV>

**<DIV ALIGN="JUSTIFY"> **

На данной странице рассматриваются основные виды прикладного программного обеспечения, которые устанавливаются на современных персональных компьютерах. Для более наглядного представления информации на страницы создана таблица, в которой прикладные программы ПК распределены по нескольким категориям.

** </DIV>**

<TABLE BORDER="3" WIDTH="100%">

<TR>

<TD ALIGN="CENTER" BGCOLOR="GOLD" COLSPAN="3">

Текстовые процессоры

```

</TD>
<TD ALIGN="CENTER" BGCOLOR="AQUA" COLSPAN="2">
<FONT SIZE="4">
Табличные процессоры
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="PINK" COLSPAN="3">
<FONT SIZE="4">
Графические редакторы
</FONT>
</TD>
</TR>
<TR>
<TD ALIGN="CENTER" BGCOLOR="GOLD">
<FONT SIZE="4">
Microsoft Word
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="GOLD">
<FONT SIZE="4">
WordPad
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="GOLD">
<FONT SIZE="4">
LibreOffice Writer
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="AQUA">
<FONT SIZE="4">
Microsoft Excel
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="AQUA">
<FONT SIZE="4">
LibreOffice Calc
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="PINK">
<FONT SIZE="4">
Adobe Photoshop
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="PINK">
<FONT SIZE="4">

```

```

Adobe Illustrator
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="PINK">
<FONT SIZE="4">
Corel Draw
</FONT>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

К первому из указанных в таблице классов «текстовые процессоры» относятся 3 перечисленные в таблице программы. Поэтому в верхней строке таблицы ячейка, в которой указан этот класс программного обеспечения, должна по ширине соответствовать 3 нижним ячейкам. Для этого в теге верхней ячейки следует установить значение атрибута **COLSPAN**, равное 3. К второму классу «табличные процессоры» в таблице относятся 2 программы, которым соответствуют 2 ячейки в нижней строке таблицы. Поэтому в теге для ячейки верхней строки таблицы, относящейся к табличным процессорам, значение атрибута **COLSPAN** равно 2. По этой же причине для класса «графические редакторы», к которому относятся 3 программы, значение атрибута **COLSPAN** в верхней ячейке равно 3. Внешний вид данной страницы показан на рис. 5.3.

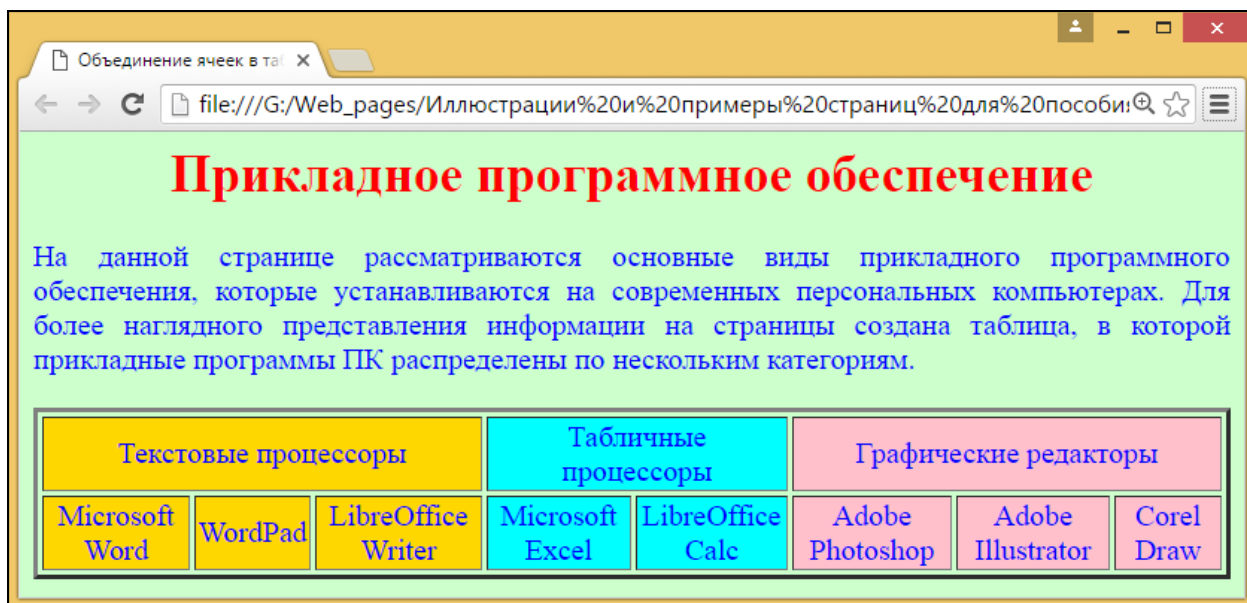


Рис. 5.3. Внешний вид страницы, содержащей таблицу, в которой ячейки объединены по горизонтали.

Помимо всех перечисленных атрибутов в таблицах можно использовать еще два атрибута, которые могут оказывать существенное влияние на ее

внешний вид: **CELLSPACING** и **CELLPADDING**. Данные атрибуты задаются для всей таблицы в целом и указываются в открывающей части тега **<TABLE>**.

Атрибут **CELLSPACING** задает расстояние между отдельными ячейками таблицы, выраженное в пикселях. Если задать значение этого атрибута равным нулю, то границы между ячейками вообще исчезнут. Если же задать значение этого атрибута больше 1 пикселя, то толщина границ между ячейками заметно увеличится.

Атрибут **CELLPADDING** задает размеры полей внутри ячеек. Эта величина также задается в пикселях. Данный атрибут задает одновременно равные значения для левого, правого, верхнего и нижнего полей.

Рассмотрим использование данных тегов на следующем примере. В таблице, приведенной на рис 5.3, в ряде ячеек надписи вплотную примыкают к границам ячеек, что ухудшает внешний вид таблицы. Добавим в тег **<TABLE>** для данной таблицы некоторые дополнительные атрибуты, не изменяя весь остальной код для этой страницы. Открывающая часть тега изменится следующим образом:

```
<TABLE BORDER="4" WIDTH="100%" CELSPACING="4"
CELLPADDING="8">
```

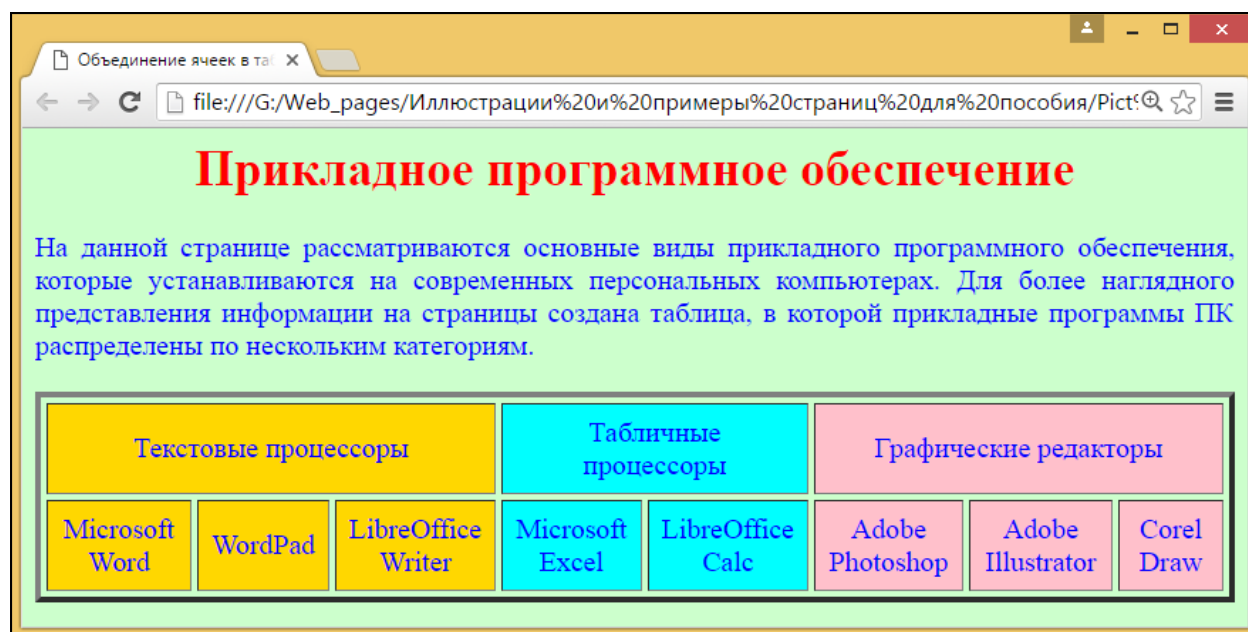


Рис. 5.4. Страница, содержащая таблицу, в которой использованы дополнительные атрибуты **CELLSPACING** и **CELLPADDING**

Внешний вид таблицы после внесенных изменений показан на рис.5.4. Видно, что использование дополнительных тегов не только улучшает оформление страницы, но и делает более удобным чтение и восприятие имеющейся в таблице информации.

5.2. Вложенные таблицы

В разделе 5.1 были рассмотрены простые таблицы, в которых в каждой ячейке содержится только текст. Но содержанием любой ячейки, имеющейся в таблице, кроме текста могут быть также элементы компьютерной графики и другие таблицы, которые являются вложенными по отношению к основной. Такая система вложений позволяет создавать достаточно сложную табличную структуру. Фактически возможна (и часто реализуется на практике) ситуация, когда вся Web-страница представляет собой одну большую таблицу, содержащую другие вложенные таблицы, причем вложение может иметь несколько уровней.

Следует также отметить, что применение вложенных таблиц может использоваться не только как способ упорядочения внутренней структуры Web-страницы, но и как способ нестандартного оформления самих страниц.

Например, в языке HTML отсутствуют штатные средства, которые можно было бы использовать для создания в таблице рамки заданного разработчиком цвета. Поэтому все рамки, которые присутствуют в примерах, приведенных в разделе 5.1, имеют одинаковый серый цвет. Но способ решения данной проблемы существует и состоит в следующем.

На Web-странице создается таблица, содержащая всего один ряд и одну ячейку. В открывающей части тега **<TABLE>** для данной таблицы с помощью атрибута **BGCOLOR** указывается фоновый цвет, соответствующий тому цвету рамки, который хочет задать разработчик. Также необходимо атрибутом **CELLPADDING** определить размеры полей для этой таблицы. Значение атрибута **CELLPADDING**, выраженное в пикселях, должно соответствовать желаемой толщине рамки создаваемой таблицы.

Затем в единственную ячейку этой таблицы вставляется еще одна таблица, которая может содержать любое необходимое разработчику количество строк и столбцов. Эта вложенная таблица займет все пространство ячейки внешней таблицы за исключением полей, которые и образуют рамку, окаймляющую данную вложенную таблицу. При этом цвет такой рамки будет определяться фоновым цветом ячейки, а ширина – соответственно шириной полей ячейки.

В качестве примера использования этого приема для оформления цветной табличной рамки возьмем таблицу из раздела 5.1, входящую в Web-страницу, изображенную на рис. 5.4, и изменим HTML-код этой страницы. Создадим новую таблицу, содержащую один ряд и одну ячейку. Для этой таблицы зададим синий фоновый цвет и величину поля, равную 3 пикселям.

Затем возьмем уже имеющуюся на странице старую таблицу и вставим ее как содержимое в единственную ячейку новой таблицы. В результате мы получим вокруг этой старой таблицы рамку синего цвета толщиной в 3 пикселя. Ниже приведен HTML-код, описывающий вышеприведенные действия по вложению таблиц друг в друга (полностью приводить HTML-код

страницы нет необходимости, так как, за исключением таблиц, он полностью идентичен предыдущему).

```
<TABLE          CELLSPACING="0"          CELLPADDING="3"
BGCOLOR="BLUE">
  <TR>
    <TD>
      <TABLE WIDTH="100%"
        CELLPADDING="8">
          <TR>
            <TD ALIGN="CENTER" BGCOLOR="GOLD" COLSPAN="3">
              <FONT SIZE="4">
                Текстовые процессоры
              </FONT>
            </TD>
            <TD ALIGN="CENTER" BGCOLOR="AQUA" COLSPAN="2">
              <FONT SIZE="4">
                Табличные процессоры
              </FONT>
            </TD>
            <TD ALIGN="CENTER" BGCOLOR="PINK" COLSPAN="3">
              <FONT SIZE="4">
                Графические редакторы
              </FONT>
            </TD>
          </TR>
          <TR>
            <TD ALIGN="CENTER" BGCOLOR="GOLD">
              <FONT SIZE="4">
                Microsoft Word
              </FONT>
            </TD>
            <TD ALIGN="CENTER" BGCOLOR="GOLD">
              <FONT SIZE="4">
                WordPad
              </FONT>
            </TD>
            <TD ALIGN="CENTER" BGCOLOR="GOLD">
              <FONT SIZE="4">
                LibreOffice Writer
              </FONT>
            </TD>
            <TD ALIGN="CENTER" BGCOLOR="AQUA">
              <FONT SIZE="4">
                Microsoft Excel
```

```

</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="AQUA">
<FONT SIZE="4">
LibreOffice Calc
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="PINK">
<FONT SIZE="4">
Adobe Photoshop
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="PINK">
<FONT SIZE="4">
Adobe Illustrator
</FONT>
</TD>
<TD ALIGN="CENTER" BGCOLOR="PINK">
<FONT SIZE="4">
Corel Draw
</FONT>
</TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>

```

На рис. 5.5 показана Web-страница после использования в ней вложенных таблиц. Теперь таблица с перечнем программ окаймлена сплошной рамкой синего цвета, созданной с помощью наружной таблицы.

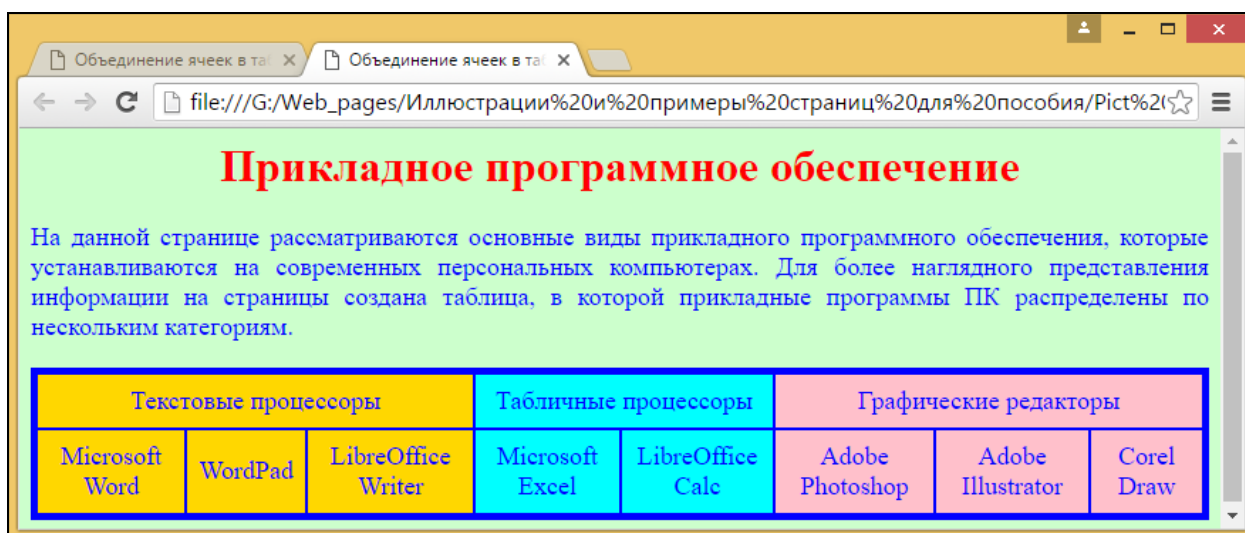


Рис. 5.5. Создание сплошной рамки с помощью вложения таблиц

Рассмотрим еще один пример создания Web-страницы на основе вложенных таблиц. Эта страница посвящена творчеству знаменитого английского писателя Джона Руэла Толкина. Большую часть данной страницы (кроме заголовка первого уровня в начале страницы, разделительной линии и одного предложения, расположенного под ней) занимает таблица, содержащая одну строку и 3 ячейки. Рамка этой таблицы явным образом не задается, поскольку в данном случае нам требуется не таблица как таковая, а средство для упорядочения контента, представленного на Web-странице.

Первая ячейка таблицы содержит обычный текст, для оформления которого используется шрифт Comic Sans MS. Поскольку в базовой таблице имеются поля шириной в 10 пикселей, этот текст непосредственно не соприкасается с левой границей страницы и с изображением, расположенным правее.

Вторая ячейка не имеет текстового наполнения и выполняет роль разделителя для текста, находящегося слева от нее и текста с таблицей, расположенного справа. Для заполнения этой ячейки используется фоновый рисунок, находящийся в файле **Ornament.jpg**. Рисунок был создан в графическом редакторе **Paint** и файл с этим рисунком был сохранен в формате **jpeg** для компактного представления данных на странице. Данный файл находится в той же папке, что и HTML-файл, и поэтому, указывая его в качестве значения атрибута **BACKGROUND**, достаточно прописать только имя файла с рисунком без указания пути к нему.

Третья ячейка имеет более сложную структуру, чем две предыдущие, поскольку она содержит как текст, так и вложенную таблицу, состоящую из 5 строк, каждая из которых в свою очередь включает 2 ячейки. Таблица окаймлена рамкой синего цвета, а фоновый цвет ячеек – белый. В данном случае для получения нужной цветовой гаммы был использован следующий прием. Для таблицы в целом был задан фоновый синий цвет, и задано с помощью атрибута **CELLSPACING** расстояние между ячейками, равное 4 пикселям. Затем для каждой строки, имеющейся в таблице, с помощью атрибута **BGCOLOR** был указан в качестве фонового белый цвет. Таким образом, пространство между ячейками, сохраняющее синий цвет, фактически образует рамку данной вложенной таблицы.

Поскольку, как уже было сказано ранее, внешняя таблица имеет поля в 10 пикселей, текст, содержащийся в третьей ячейке, и вложенная в нее таблица не будут соприкасаться с правой границей страницы.

Полный HTML-код данной Web-страницы приводится ниже:

```
<HTML>
<HEAD>
<TITLE> Создание вложенных таблиц </TITLE>
</HEAD>
<BODY TEXT="BLUE" BGCOLOR="CCFFCC">
<DIV ALIGN="CENTER">
```

```

<H1>
<FONT COLOR="RED" FACE="IMPACT">
Мир Джона Руэла Толкина
</FONT>
</DIV>
</H1>
<HR COLOR="MAROON" SIZE="4">
<FONT SIZE="5" COLOR="GREEN">
<B> <I>
Среднеземье - фэнтезийный мир, созданный знаменитым
английским писателем Джоном Рональдом Руэлом Толкиным
</I></B>
</FONT>
<TABLE          WIDTH="100%"          CELLSPACING="5"
CELLPADDING="10" >
<TR>
<TD  WIDTH="40%" VALIGN="TOP">
<DIV ALIGN="JUSTIFY">
<FONT FACE="COMIC SANS MS" SIZE="4">
<B>
&nbsp;
История и география созданного писателем мира
Среднеземья подробно описаны в знаменитой трилогии
"Властелин колец", которая включает в себя романы
"Братство кольца", "Две башни", "Возвращение короля". В
трилогии рассказывается о грандиозной войне между
силами добра и зла, происходившей в этом мире много
тысяч лет назад.
<BR>
&nbsp;
Сведения о Среднеземье и его обитателях, о
предыстории событий, о которых повествует трилогия,
можно также почерпнуть из таких произведений писателя,
как "Хоббит" и "Сильмариллион"
</B>
</FONT>
</DIV>
</TD>
<TD WIDTH="20%" BACKGROUND="Ornament.jpg">
</TD>
<TD VALIGN="TOP">
<DIV ALIGN="JUSTIFY">
<FONT FACE="COMIC SANS MS" SIZE="4">
<B>
&nbsp;

```

В приведенной ниже таблице указывается примерная численность представителей различных рас, населяющих Среднеземье

</DIV>

<TABLE BGCOLOR="BLUE" CELLSPACING="4">

<TR BGCOLOR="WHITE">

<TD WIDTH="200" HEIGHT="50">

Название расы

</TD>

<TD WIDTH="200">

Примерная численность

</TD>

</TR>

<TR BGCOLOR="WHITE">

<TD WIDTH="200" HEIGHT="50">

Хоббиты

</TD>

<TD WIDTH="200">

50000

</TD>

</TR>

<TR BGCOLOR="WHITE">

<TD WIDTH="200" HEIGHT="50">

Люди


```

</FONT>
</TD>
<TD WIDTH="200">
<FONT FACE="COMIC SANS MS" SIZE="4">
<B>
300000
</B>
</FONT>
</TD>
</TR>
<TR BGCOLOR="WHITE">
<TD WIDTH="200" HEIGHT="50">
<FONT FACE="COMIC SANS MS" SIZE="4">
<B>
Эльфы
</B>
</FONT>
</TD>
<TD WIDTH="200">
<FONT FACE="COMIC SANS MS" SIZE="4">
<B>
250000
</B>
</FONT>
</TD>
</TR>
<TR BGCOLOR="WHITE">
<TD WIDTH="200" HEIGHT="50">
<FONT FACE="COMIC SANS MS" SIZE="4">
<B>
Орки
</B>
</FONT>
</TD>
<TD WIDTH="200">
<FONT FACE="COMIC SANS MS" SIZE="4">
<B>
500000
</B>
</FONT>
</TD>
</TR>
</TABLE>
</TD>
</TR>

```

```

</TABLE>
</BODY>
</HTML>

```

Внешний вид страницы, созданной с помощью данного кода, показан на рис.5.6. Как видно на приведенной иллюстрации, использование вышеописанных, сравнительно несложных, приемов позволяет придать странице достаточно нестандартное оформление.

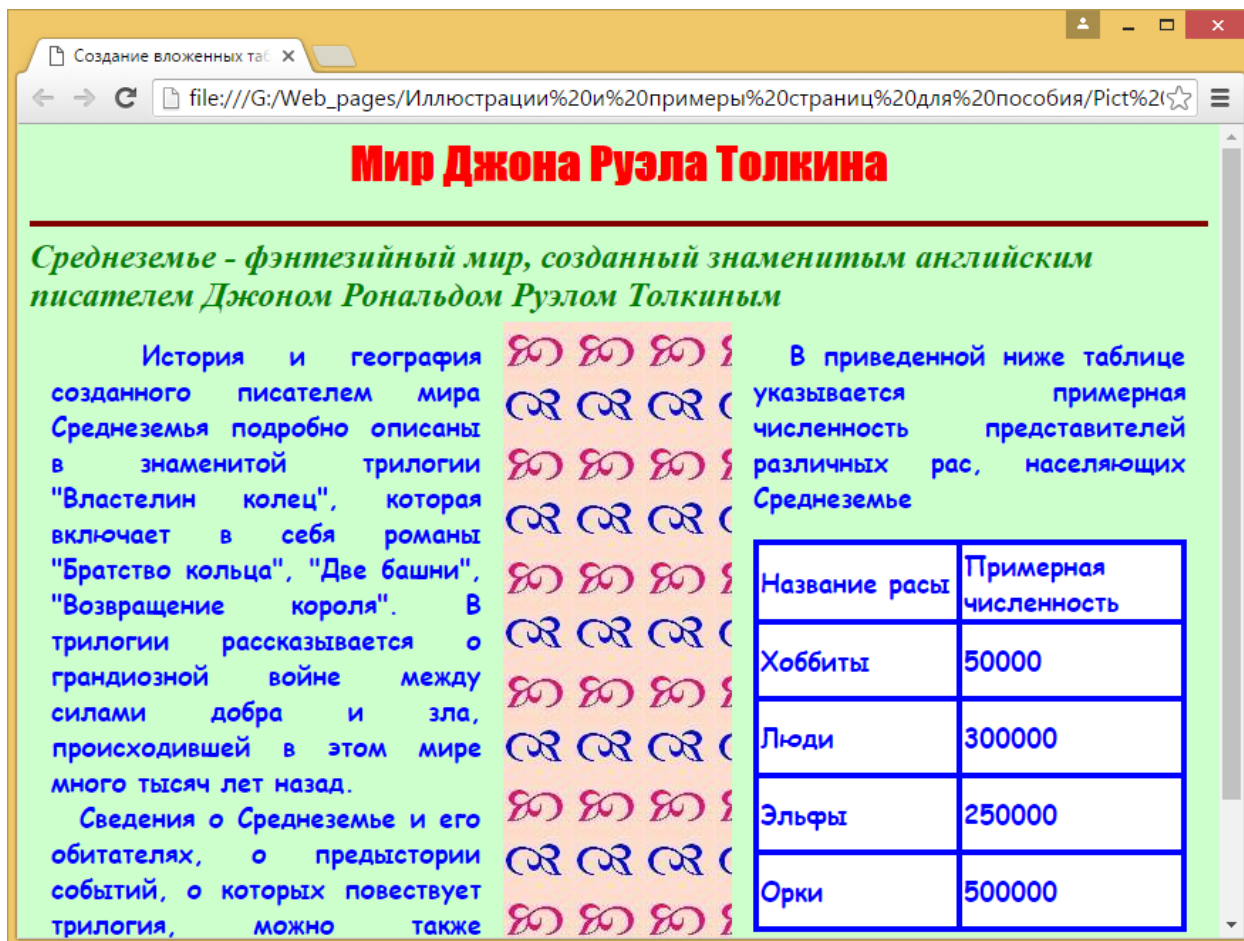


Рис. 5.6. Пример создания Web-страницы на основе вложенных таблиц

Вопросы для самоконтроля

1. Для чего используются таблицы в Web-страницах?
2. С помощью какого тега создаются таблицы?
3. Какие атрибуты позволяют определить количество строк и столбцов в создаваемой таблице?
4. Каким образом можно регулировать ширину ячеек в таблице?
5. Как можно объединить в таблице несколько ячеек?
6. Какие атрибуты позволяют выполнять выравнивание текста в ячейках по горизонтали?
7. Какие атрибуты позволяют выравнивать текст в ячейках по вертикали?
8. Как задается расстояние между текстом, находящимся в ячейке, и границей ячейки?
9. Как можно отрегулировать толщину рамки для таблицы?
10. Как можно задать определенный фоновый цвет ячейки?
11. Каким образом можно использовать рисунок в качестве фона ячейки?
12. Каким образом можно задать цвет рамки таблицы?

Библиографический список

- Дронов В.А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. – СПб.: БХВ-Петербург, 2011.
- Квинт И. HTML, XHTML и CSS на 100 %. – СПб.: Питер, 2010.
- Ломов А.Ю. HTML, CSS, скрипты: практика создания сайтов. – СПб.: БХВ-Петербург, 2006.
- Мержевич В.В. HTML и CSS на примерах. – СПб.: БХВ-Петербург, 2005.
- Прохоренок Н.А. HTML, Javascript, PHP и MySQL. – 3-е изд. – СПб.: БХВ-Петербург, 2010.
- Симонович С.В. Информатика. Базовый курс: учебник для вузов. – 3-е изд. – СПб.: Питер, 2014.

Учебное издание

Артур Николаевич Маслобоев

**БАЗОВЫЕ ВОЗМОЖНОСТИ ЯЗЫКА HTML
ПО РАЗРАБОТКЕ И ОФОРМЛЕНИЮ
WEB-СТРАНИЦ**

Учебно-методическое пособие

Редактор и корректор Н.П. Новикова

Техн. редактор Л.Я. Титова

Темплан 2016 г., поз. 53

Подп. к печати 01.06.16 Формат 60x84/16. Бумага тип. № 1.
Печать офсетная. 5,5 уч.-изд. л.; 5,5 усл. печ. л. Тираж 50 экз. Изд. № 53.
Цена «С». Заказ

Ризограф Высшей школы технологии и энергетики СПбГУПТД, 198095,
СПб., ул. Ивана Черных, 4.