

Глава 1

Первые шаги в HTML

В этой главе...

- ◆ Введение. Инструменты
- ◆ Создаем HTML-документ
- ◆ Цвет и фон
- ◆ Форматирование текста
- ◆ Списки
- ◆ Резюме

Введение. Инструменты

Прежде чем приступить к занятиям, давайте разберемся, что же такое язык HTML и для чего он нужен? HTML (HyperText Markup Language — язык гипертекстовой разметки) предназначен для разметки и оформления документов, публикуемых в World Wide Web (WWW) или, проще сказать, HTML-документов. Создатели позаботились о том, чтобы этот язык был независимым от платформы, т.е. мог работать в любых операционных средах. Основными элементами языка HTML стали дескрипторы (или тэги, tags) — операторы, названия которых заключаются в угловые скобки. Документы, размеченные при помощи этого языка, визуализируются браузерами¹ конечных пользователей в большинстве случаев одинаково благодаря тому, что “понимают” и правильно обрабатывают структурные элементы языка HTML. Исходный код представляет собой текст, отформатированный с помощью дескрипторов, причем посетителю Web-страницы эти элементы не видны, а виден лишь результат их воздействия на документ.

На сегодня существует множество редакторов, которые помогают программисту в верстке HTML-страниц. Кстати, не помешает сказать, что HTML-документ — это не более чем текстовый файл с расширением *.html или *.htm. Создатели многих HTML-редакторов позаботились в своих программных продуктах об удобной правке страниц, легкой вставке в них графических файлов — и главное, о том, чтобы результат работы можно было бы сразу увидеть на экране монитора. Однако используя такого рода редакторы, об оптимизации кода можно забыть. Визуальный, или правильнее сказать WYSIWYG-редактор (What You See Is What You Get — что вы видите, то и получаете), а мы говорили именно о таких редакторах, при создании HTML-файла вписывает в него много “мусора” (повторяющиеся дескрипторы и т.п.), который в большинстве случаев лишь затрудняет загрузку страницы. Происходит это из-за того, что программа сама определяет, какими дескрипторами ей нужно форматировать документ, и иногда выбирает совсем не то, что надо автору. Это, наверное, самый

¹ Браузер — от английского слова *Browser* (окно просмотра или обозреватель). В данном контексте имеется в виду программа-обозреватель *Internet* — *Internet Explorer*, *Opera*, *Netscape Navigator* или любая другая.

большой минус визуальных HTML-редакторов, но мы не будем на них останавливаться, а уж рассказывать о том, как с ними работать, и подавно. Нас в действительности интересуют редакторы другой группы — напрямую работающие с HTML-кодом. К таким программам, в частности, можно отнести стандартное приложение “Блокнот” Windows. Да-да, обычный Блокнот, в котором многие пользователи привыкли создавать текстовые файлы. И не слушайте людей, которые говорят, что, если вы ничего не понимаете в HTML, то для начала вам больше подойдут визуальные редакторы. Ничего подобного, так вы ничему не научитесь. Откройте созданный визуальным редактором документ в Блокноте — и вам покажется, что вы находитесь ночью в непроходимых джунглях. Конечно, если вы работаете на какой-нибудь фирме и поддерживаете ее сайт, то тут нечего особо напрягаться и можно применить визуальный редактор, а когда делаешь сайт для себя — тут уже совсем другие пироги ... Я не призываю относиться халатно к работе, но согласитесь, что делать для кого-то и для себя — вещи совершенно разные.

На данный момент не так уж и много программ, работающих с HTML-кодом напрямую, и хотя в процессе работы к этим программам мы будем обращаться нечасто, упомянуть о них все же стоит. Очень неплохая программка из этой области — HTMLPad. В ней есть подсветка синтаксиса (это когда определенные элементы подсвечиваются другим цветом, отличающим их от основного текста), окно просмотра результата, база специальных символов и прочие полезные функции, доступ к которым очень прост: достаточно взглянуть на панель инструментов. Внешний вид окна программы HTMLPad представлен на рис. 1.1. Поскольку данное программное обеспечение платное, вам придется немного раскошелиться на его покупку, или же работать с ним лишь определенный промежуток времени, который программисты установили для бесплатной эксплуатации в целях ознакомления. Эта программа пусть будет у вас, как говорится, про запас, к ней можно будет обращаться по мере надобности.

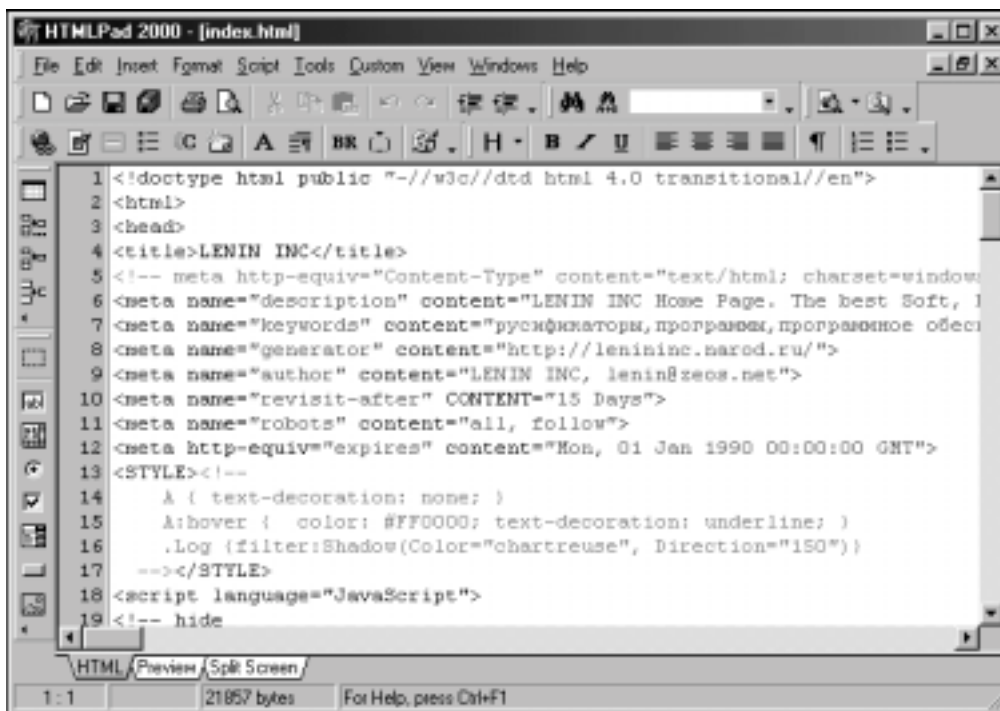


Рис. 1.1. Внешний вид окна программы HTMLPad

В первую же очередь нам понадобится простенький текстовый редактор, напоминающий Блокнот, — Vred, который распространяется совершенно бесплатно и уже успел приобрести немало почитателей. На первый взгляд Vred ни чем не отличается от стандартного Блокнота, однако первое впечатление всегда обманчиво.

В отличие от Блокнота, эта программа содержит множество разных настроек: подсветку HTML-дескрипторов, автоматическую замену текста или символов, регистрацию расширений (*.txt, *.log, *.scp, *.ini, *.inf), есть панели инструментов и статуса, реализована неограниченная отмена действий (ограничения накладываются только объемом доступной памяти), а лимит текста — не менее одного мегабайта. Здесь перечислены далеко не все возможности этого “простенького” текстового редактора... Внешний вид окна программы представлен на рис. 1.2. Рекомендуется заменить стандартный Блокнот Windows программой Vred. Если вы решитесь на подобную замену, то делается это так: переименуйте исполняемый файл программы Блокнот — файл notepad.exe как файл копии, — например, notepad.bak (удалять не будем), а исполняемый файл программы Vred переименуйте в notepad.exe.

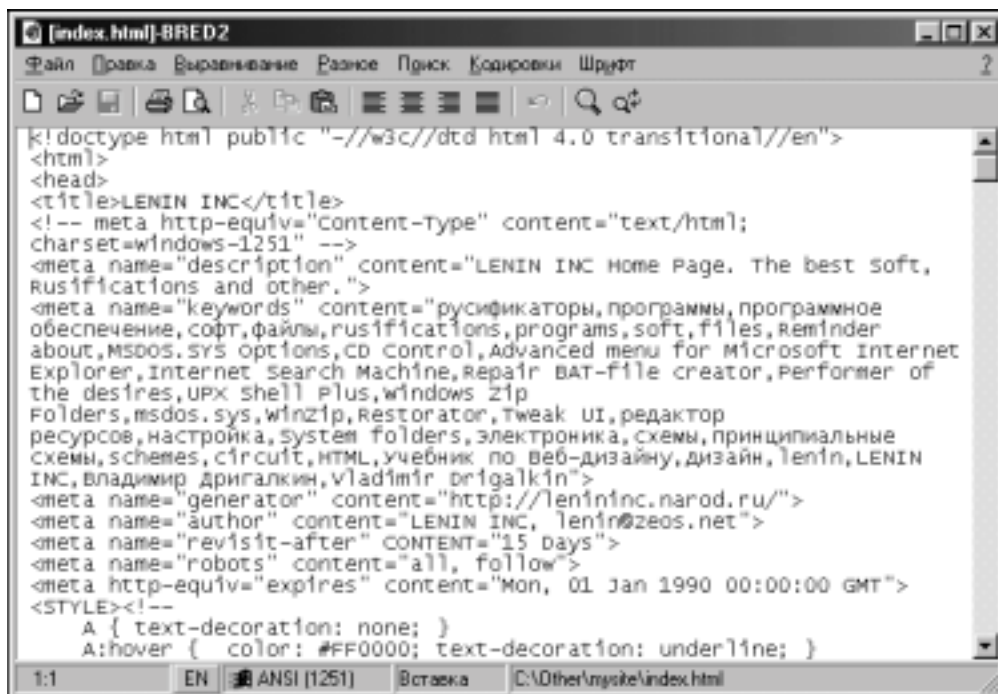


Рис. 1.2. Внешний вид окна программы Vred

Кроме этого, чтобы работа с HTML-документами была еще удобнее, добавьте в контекстное меню Windows пункт Открыть в Блокноте. Для этого необходимо создать текстовый файл и поместить в него следующие команды:

REGEDIT4

```
[HKEY_CLASSES_ROOT*\Shell\Открыть в Блокноте\command]
@="notepad.exe %1"
```

Сохраните этот файл с именем, например, notepad.reg и запустите его на выполнение; на вопрос системы, следует ли добавить информацию в реестр, ответьте положительно. В результате соответствующие данные будут внесены в реестр Windows, и в контекстном меню появится пункт Открыть в Блокноте, как показано на рис. 1.3.



Рис. 1.3. Новый пункт Открыть в Блокноте в контекстном меню Windows

Теперь после щелчка правой кнопкой мыши на любом из файлов в окне какой-либо папки, в раскрывшемся контекстном меню будет присутствовать команда Открыть в Блокноте. Это очень удобно во многих ситуациях, например, когда расширение текстового файла изменено на *.diz.

На этом предварительный обзор HTML-редакторов и обсуждение вопросов подготовки к предстоящей работе можно завершить.

Допустим, что вы установили программы, описанные выше, проделали все эти изменения и наслаждаетесь нововведениями. Но рано или поздно неизбежно появится еще один важный вопрос: в каком графическом редакторе лучше всего создавать изображения для Web-страниц? Я не буду рассказывать, какие графические редакторы плохие и почему, скажу только, что самый лучший, на мой взгляд, это — Paint Shop Pro. Ни один существующий на данный момент редактор графики не может тягаться с ним по скорости, качеству сохранения изображений и степени их оптимизации. К сожалению, данный редактор поддерживает только английский язык, но на сайтах русификации можно поискать русификатор к этой программе. Кстати, за ее регистрацию вам также придется заплатить. Внешний вид окна этой популярной программы представлен на рис. 1.4.

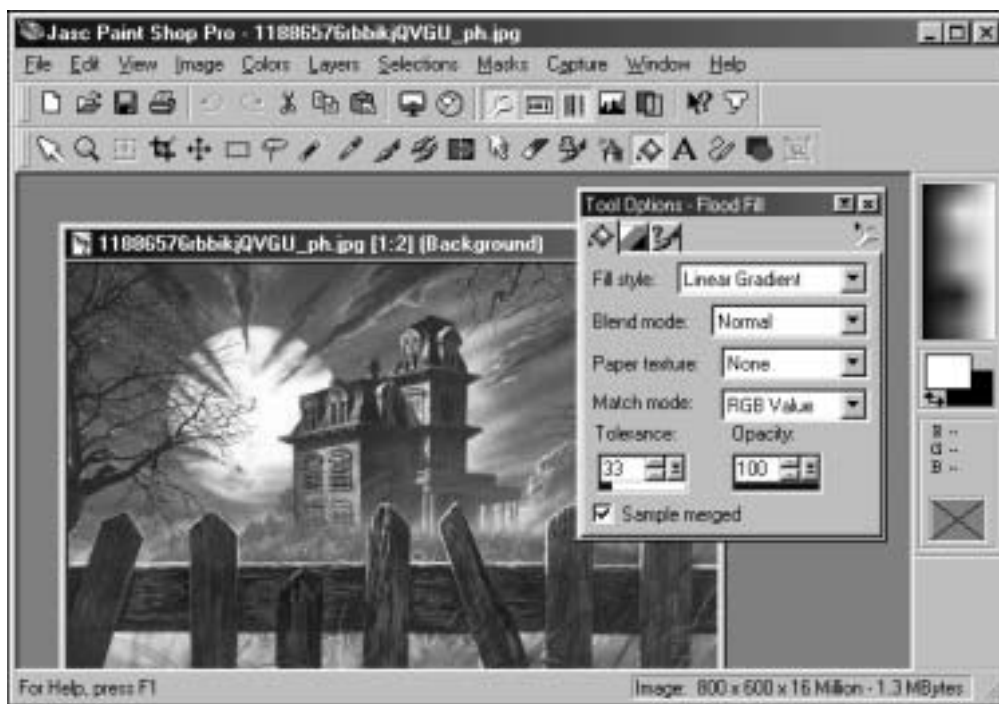


Рис. 1.4. Общий вид окна программы Paint Shop Pro

Если ваш будущий сайт имеет радиотехническое направление, и вам потребуется рисовать принципиальные схемы, воспользуйтесь для этой цели программой CircuitMaker. К сожалению, русским интерфейсом и бесплатностью этой программы я вас также не смогу порадовать, однако опытный радиолюбитель вполне сможет разобраться в ней самостоятельно. Общий вид окна этой программы представлен на рис. 1.5.

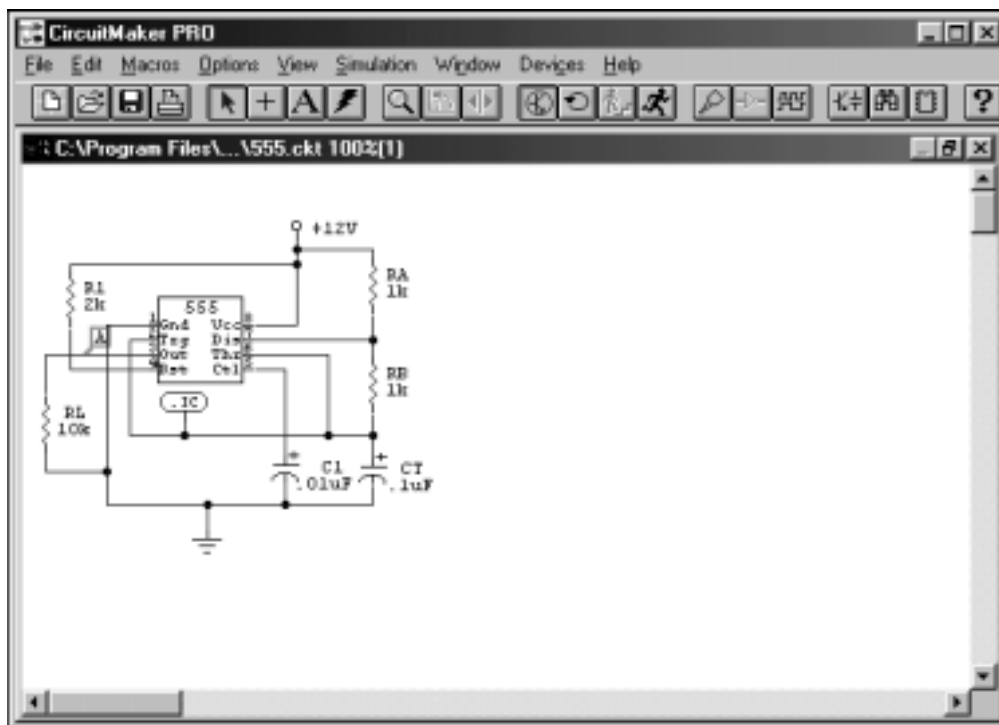


Рис. 1.5. Общий вид окна программы CircuitMaker

Последнее время большой популярностью пользуются различные Flash-технологии. Программисты все чаще применяют их для оформления своих сайтов. Flash — это своего рода анимация или точнее видеоролик. Здесь есть как множество своих плюсов (например, маленький размер файлов), так и множество минусов (загруженность HTML-страницы при неправильном использовании Flash). Создает подобную анимацию программа Flash, общий вид главного окна которой представлен на рис. 1.6. Своим дизайном она немногим отличается от любого популярного графического редактора: та же панель инструментов с кнопками выбора кистей, ластика, карандаша и прочих инструментов. Однако в этом редакторе мы будем создавать анимационные ролики, меню к своему сайту, заставки, баннеры² и прочие различные “украшательства”.

В принципе, на этом можно окончательно завершить обзор тех инструментов Web-программиста, которые потребуются ему в первую очередь, и перейти к долгожданному практическим занятиям.

Все описанные здесь программы вы сможете найти по Web-адресам, указанным ниже.

Программа	Статус	Ссылка
Bred	Бесплатная	http://www.astonshell.com/rus/
Paint Shop Pro	Платная	http://www.jasc.com/
CircuitMaker	Платная	http://www.microcode.com/
Flash	Платная	http://www.macromedia.com/software/flash/

² Баннер — изображение-ссылка рекламного характера.

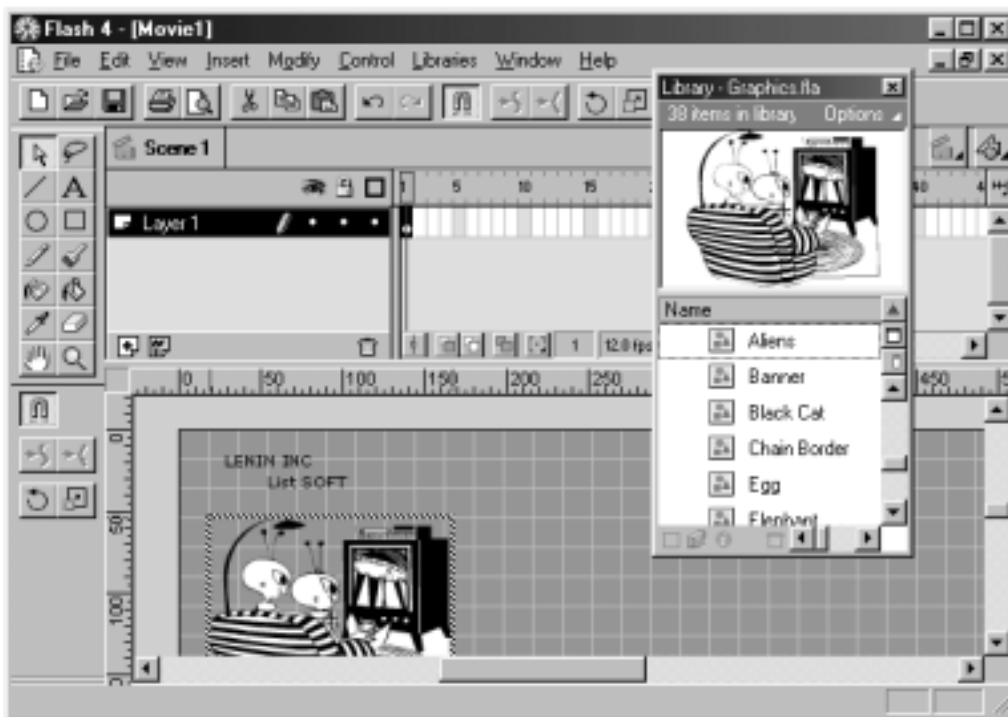


Рис. 1.6. Общий вид окна программы Flash

Создаем HTML-документ

Итак, вы еще не знаете языка HTML, и даже не сделали ничего из того, что было предложено в первой главе: по каким-то причинам не установили ни одну из программ, но у вас есть огромное желание создать свою Web-страничку, а это уже немало. Что ж, попробуем построить Web-сайт, используя стандартный Блокнот, и ознакомимся с некоторыми основными (базовыми) дескрипторами языка HTML, без которых не может обойтись любая Web-страница.

Откройте приложение Блокнот (или редактор Вред, если вы его все же установили) и наберите в нем текст, представленный в следующем абзаце. При этом то, что выделено курсивом, набирать не обязательно — это комментарии, которые игнорируются браузером, но иногда очень помогают программисту не путаться в собственном же HTML-документе. Все, что выделено полужирным шрифтом, вы оставляете так, как есть, а обычный текст исправляете на свой или же пока не изменяете.

```

<!--Пример HTML-документа.-->
<html>
<head>
<!--Следующая строка - кодировка.-->
<meta http-equiv="Content-Type"
  content="text/html; charset=windows-1251">
<!--Заголовок HTML-документа.-->
<title>Заголовок</title>
</head>
<!--Тело HTML-документа.-->

```

```

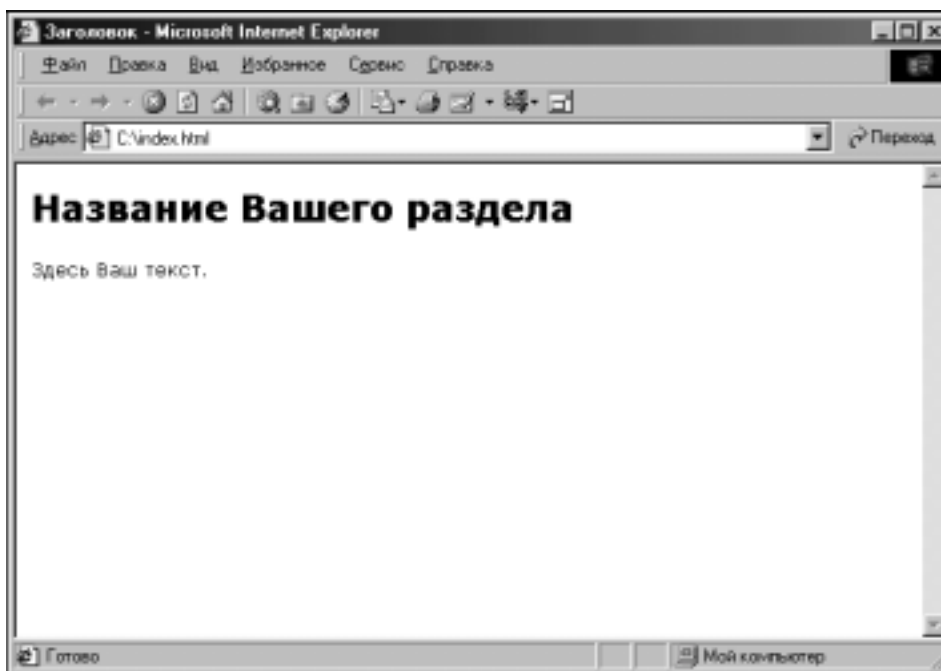
<body>
<!--Теперь указываем название шрифта,
его размер и цвет.-->
<font face="Verdana" size="2" color="#000000">
<h2> Название вашего раздела </h2>
Здесь ваш текст.
<!--Встраиваем графический файл в документ.
И будет он у нас по центру.-->
<p align="center">

</p>
</font>
</body>
</html>

```

Сохраните документ как .html-файл (например, под именем index.html¹) и запустите его на выполнение. Вы создали свою первую страницу. Примите мои поздравления!

Ничего, что она выглядит просто, потом, со временем и приходящим опытом, вы будете ее дополнять и улучшать, а пока ограничимся самым необходимым. На рис. 1.7 показано, как должна выглядеть наша страничка. Изображения в нем пока нет, ведь мы еще не знаем, как вставить его в HTML-документ, но далее будет рассказано и об этом.



¹ index.html — это стандартное имя входной Web-страницы, т.е. когда вы в своем браузере вводите адрес, например, <http://lenininc.narod.ru/>, на сервере ищется файл index.html. Если его по указанному адресу нет, то выдается ошибка, что документ не найден, а если есть — вы автоматически попадаете на первую страницу данного Web-сайта.

Рис. 1.7. Пример HTML-документа

Рассмотрим дескрипторы, которые были использованы в данном примере. Большинство дескрипторов имеют два варианта написания — как открывающий элемент `<>` и как закрывающий элемент `</>`. Между ними находятся коды, распознаваемые браузером. Некоторые дескрипторы закрывать (т.е. записывать их закрывающий вариант) обязательно, другие — не обязательно.

HTML-документ всегда должен начинаться открывающим дескриптором `<HTML>` и заканчиваться закрывающим дескриптором `</HTML>`. Далее обычно следует два дескриптора заголовка `<HEAD>` и `</HEAD>`. Между этими дескрипторами обязательно должна находиться информация о кодировке² данной Web-страницы, в нашем случае она выглядит так, как показано ниже.

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

Это означает, что используется кодовая страница с номером 1251 — Кириллица (Windows). Далее обычно следует пара дескрипторов, определяющих название страницы `<TITLE>` — `</TITLE>`. И только теперь, между парой дескрипторов содержимого документа `<BODY>` — `</BODY>`³ можно поместить собственную информацию, рисунки, видео- и аудиофайлы. При этом регистр букв в именах дескрипторов совершенно не важен, т.е. значение `<body>` полностью эквивалентно значению `<BODY>`, тогда как при указании в параметрах дескрипторов имен файлов регистр букв имеет важное значение. Если используемый файл называется `images.jpg`, а в HTML-коде вы его описали как `Images.jpg`, или `IMAGES.JPG`, или `images.JPG`, он просто окажется не найден на сервере и, соответственно, при открытии страницы его содержимое не отобразится в браузере пользователя. Все значения, присваиваемые отдельным параметрам дескрипторов, правильнее будет заключать в кавычки: `align="center"`, но это требование не является обязательным.

Более подробную информацию о существующих дескрипторах языка HTML можно найти в приложении А “Элементы языка HTML”.

Наконец-то мы дошли до изображения. Думаю, у вас не возникнет трудностей с его вставкой, так как здесь все крайне просто. Параметр `SRC` указывает имя графического файла, который должен быть в одном из следующих форматов: `.jpg`, `.gif` или `.png`. Размеры изображения (в пикселях) этого графического файла (параметры `WIDTH` и `HEIGHT`), а также текст всплывающей подсказки (параметр `ALT`) можно не указывать, но тогда страничка будет грузиться дольше, не забывайте об этом. Кроме того, большие изображения браузер может игнорировать, если не будут указаны их размеры. Размер конкретного изображения можно посмотреть в любом графическом редакторе или программе просмотра графических файлов, даже в старой доброй программе Paint, поставляемой вместе с Windows (команда Рисунок⇒Атрибуты).

Итак, с изображениями мы разобрались и даже исправили значения всех параметров дескриптора на требуемые (например, так, как показано ниже).

```

```

Теперь наш HTML-документ будет включать указанное изображение, как показано на рис. 1.8.

² Кодировка ставится не во всех случаях. Некоторые серверы делают это сами, поэтому при выборе места под сайт внимательно читайте документацию по его размещению. Кодировки бывают самыми разными, но наибольшей популярностью пользуются только две — Кириллица (Windows) и Кириллица (КОИ8-Р). Как правило, первую кодировку Вы можете использовать при создании сайтов в Блокноте, последнюю же обычно применяют визуальные редакторы.

³ BODY — англ. тело, в языке HTML это понимается как основное тело документа.

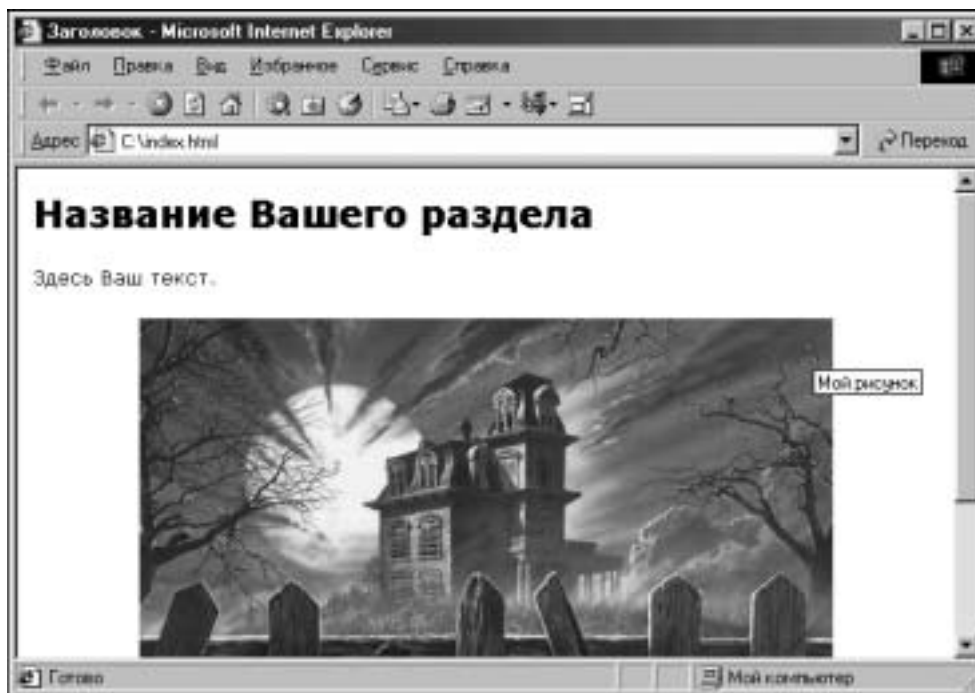


Рис. 1.8. Пример HTML-документа с изображением

Я не сомневаюсь, что, прочитав этот раздел, вы научитесь создавать HTML-документы, вставлять в них графические файлы и запомните базовые дескрипторы, без которых не обходится ни один Web-сайт, однако далее вам предстоит узнать еще много интересного.

Кстати, на основе приведенного выше HTML-кода можно создать своего рода “болванку” или заготовку, с помощью которой вы в дальнейшем сможете строить все страницы своего сайта. Это позволит на всех страницах придерживаться определенного, выбранного в самом начале стиля, что нравится пользователям и вообще выглядит очень привлекательно, поскольку все странички Web-сайта в этом случае имеют одинаковый общий вид. Если же появится необходимость внести какие-либо дополнения или изменения в уже существующие страницы, их HTML-код будет достаточно легко читаться. Полагаю, что именно в этой главе следует сказать о тех параметрах дескриптора <BODY>, с помощью которых устанавливаются размеры полей во всем документе. Они задаются числовыми значениями и имеют следующие имена:

- leftmargin — левое поле;
- rightmargin — правое поле;
- topmargin — верхнее поле;
- bottommargin — нижнее поле.

Например, в приведенном ниже дескрипторе размеры полей устанавливаются равными нулю на всех ее краях.

```
<BODY leftmargin="0" rightmargin="0"
topmargin="0" bottommargin="0">
```

Указанные параметры отлично работают в браузере Internet Explorer. Но будем предусмотрительны и откроем на всякий случай руководство по языку HTML, прилагаемое к браузеру Netscape Navigator. Здесь мы с удивлением читаем, что параметры задания размеров полей документа выглядят следующим образом:

`marginwidth` — поле слева и справа;
`marginheight` — поле сверху и снизу.

Как же быть, спросите вы? Выход есть и он очень прост — напишем HTML-код, в котором укажем размеры полей слева и сверху, этого будет достаточно, причем подобный код будет корректно работать в обоих браузерах.

```
<BODY leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
```

В этом случае Internet Explorer “видит” только свои параметры, а Netscape Navigator — свои.

В дескрипторе `<BODY>` могут также присутствовать и другие параметры, например, задающие некоторые функции языка JavaScript, которые определяют базовые цвета текста и ссылки. Подробнее об этом будет рассказано далее.

Цвет и фон

Цвет и фон (имеется в виду фоновое изображение) страницы выбираются по желанию Web-мастера, но нужно учитывать и тот факт, что от них зависит визуальное восприятие всего сайта. Не стремитесь все чрезмерно затемнить, текст удобно читать, если его хорошо видно. Не применяйте черный цвет, мотивируя это продлением срока службы мониторов пользователей. По статистическим данным посещаемость “черных” сайтов не очень высокая. Многие, скажем, корифеи Internet, которые просмотрели немало сайтов и в сети уже не один год, стараются не задерживаться на таких Web-страницах, а иногда даже не смотрят на их содержание. Вызвано это в определенной степени тем, что черный цвет зачастую используют сайты порнографической или фашистской тематики, а также сатанинские секты.

Итак, после небольшого отступления, рассмотрим сначала способ задания фонового цвета Web-страницы. Для этого в строку с дескриптором `<BODY>` создаваемого HTML-документа нужно добавить параметр `BGCOLOR` и присвоить ему требуемое значение — название цвета на английском языке, например, `red` (красный), `green` (зеленый) и т.д., или же шестнадцатеричный код желаемого цвета — `#FF0000` (красный), `#008000` (зеленый) и т.д. Браузер одинаково распознает оба варианта. Шестнадцатеричный код того или иного цвета можно посмотреть в некоторых специальных программах или графических редакторах, например в Paint Shop Pro. Кроме того, данная книга включает приложение Б “Названия цветов и их шестнадцатеричные коды”, где есть сведения о всех цветах, поддерживаемых в языке HTML. В приведенном ниже примере страница документа заполняется красным цветом и для его описания использовано название цвета.

```
<BODY BGCOLOR="Red">
```

Подобным образом можно воспользоваться и шестнадцатеричным кодом цвета.

```
<BODY BGCOLOR="#FF0000">
```

Здесь все просто, поэтому мы не будем задерживаться и терять драгоценное время на пустяки, а сразу перейдем к рассмотрению способов создания фонового изображения. Фон может быть как достаточно большой графический файл (учтите — в этом случае страничка будет грузиться дольше), так и его фрагмент, что более разумно из соображений времени загрузки. При использовании фрагмента файла он будет автоматически размножен браузером на все отображаемое поле, поэтому подбирайте такой фрагмент изображения, который хорошо стыкуется по краям с самим собой.

Вставить фоновое изображение в HTML-страницу можно с помощью следующей конструкции.

```
<BODY BACKGROUND="images.gif">
```

Здесь параметру `BACKGROUND` присвоено значение `images.gif` — это имя графического файла с расширением `.gif`. Естественно, имя файла может быть любым. Пред-

полагается, что графический файл расположен в одном каталоге с данным HTML-документом, иначе потребуется указать к нему путь. Разберемся со способами указания пути на примере файла `images.gif`.

- На два уровня каталогов вниз:
`BACKGROUND="Folder1/Folder2/images.gif"`
- На два уровня каталогов вверх:
`BACKGROUND="../../images.gif"`
- Полный путь (URL-адрес)
`BACKGROUND="http://lenininc.narod.ru/images.gif"`

В большинстве случаев лучше всего использовать полный путь, чтобы избежать путаницы. Подобное указание путей применяется для разных элементов (например, ссылок).

Что ж, с теорией мы немного ознакомились, теперь — дело за практикой. Попробуем заполнить нашу страничку фоновым изображением, показанным на рис. 1.9. Главное, чтобы выбранный фоновый рисунок был относительно бледным, не привлекал особого внимания пользователя и не мешал читать текст.

Воспользовавшись этим графическим файлом для построения фонового изображения, мы получим результат, показанный на рис. 1.10.

Существует еще один параметр, который может использоваться вместе с параметром `BACKGROUND`, это — `BGPROPERTIES`. Присвоив этому параметру значение `fixed`, можно сделать фоновое изображение неподвижным, т.е. при прокручивании страницы фон будет оставаться на месте.



Рис. 1.9. Исходное изображение для фона

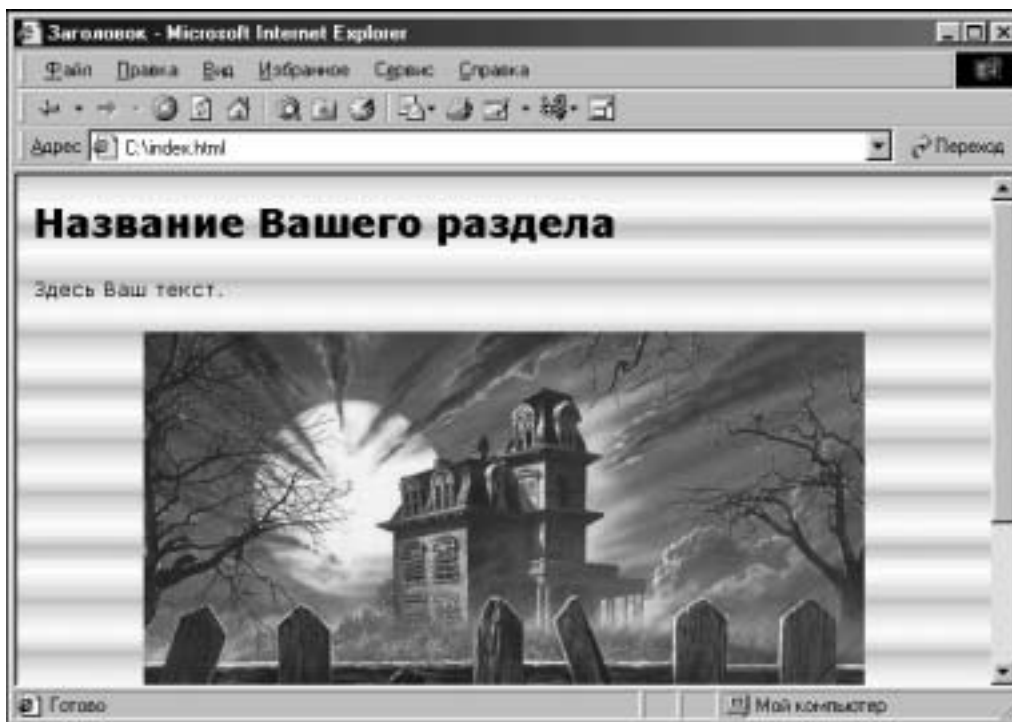


Рис. 1.10. Вид Web-страницы, полученный в результате применения фонового изображения

```
<BODY background="http://lenininc.narod.ru/
images.gif" bgproperties="fixed">
```

Любой Web-программист вам скажет, что использование фонового цвета Web-страниц более рационально из соображений быстрой загрузки сайта. Рисунки вообще грузятся дольше, даже самые маленькие. Впрочем, это ваше дело и вы можете поступать так, как считаете нужным. К тому же в последнее время появляется все больше пользователей, имеющих достаточно хорошие компьютеры, подключенные к Internet через выделенные скоростные линии. Однако не следует забывать и о скромных машинах, которых тоже пока немало, медленную модемную связь, плохие телефонные линии, не говоря уже о наших провайдерах¹.

Форматирование текста

Практически любой HTML-документ начинается с заголовка или названия статьи. Из накопленного опыта естественно будет предположить, что заголовки тоже форматировуются специальными дескрипторами языка HTML. Это действительно так — для описания заголовков используются парные дескрипторы `<Hn> </Hn>`, где *n* — число от 1 до 6, определяющее уровень данного заголовка. Заклучив между этими дескрипторами некоторый текст, мы получим соответствующий заголовок указанного уровня (рис. 1.11).

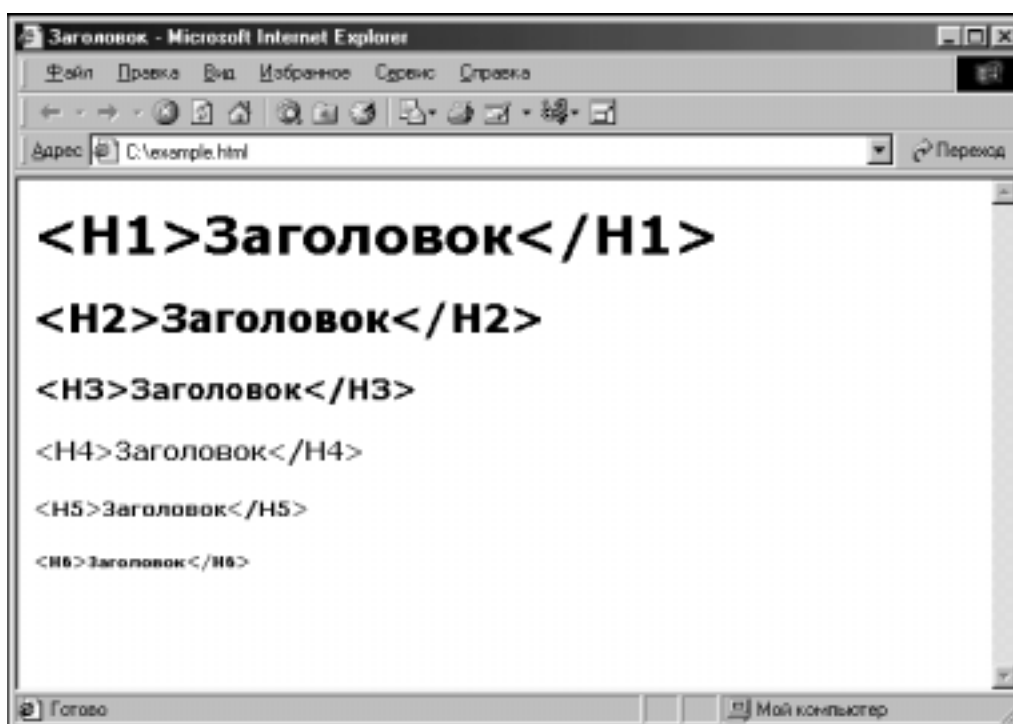


Рис. 1.11. Пример форматирования заголовков

Применение в документах дескрипторов `<Hn>` крайне желательно, так как многие поисковые серверы отыскивают в Web-документах и считывают содержимое именно

¹ Провайдер — поставщик услуг Internet.

этих дескрипторов, что способствует популяризации и упрощает поиск вашей информации потенциальными пользователями.

Для создания нового абзаца используется дескриптор <P>, а для перехода на новую строчку без создания абзаца — дескриптор
. Эти дескрипторы закрывать не обязательно; если только в дескрипторе <P> не используется элемент ALIGN, которым можно задавать выравнивание абзаца. Что это означает, можно увидеть на рис. 1.12, где показаны результаты присвоения параметру ALIGN различных значений, поясняемых ниже.

- Left — выравнивание текста по левому краю.
- Center — выравнивание текста по центру.
- Right — выравнивание текста по правому краю.
- Justify — выравнивание текста по ширине.

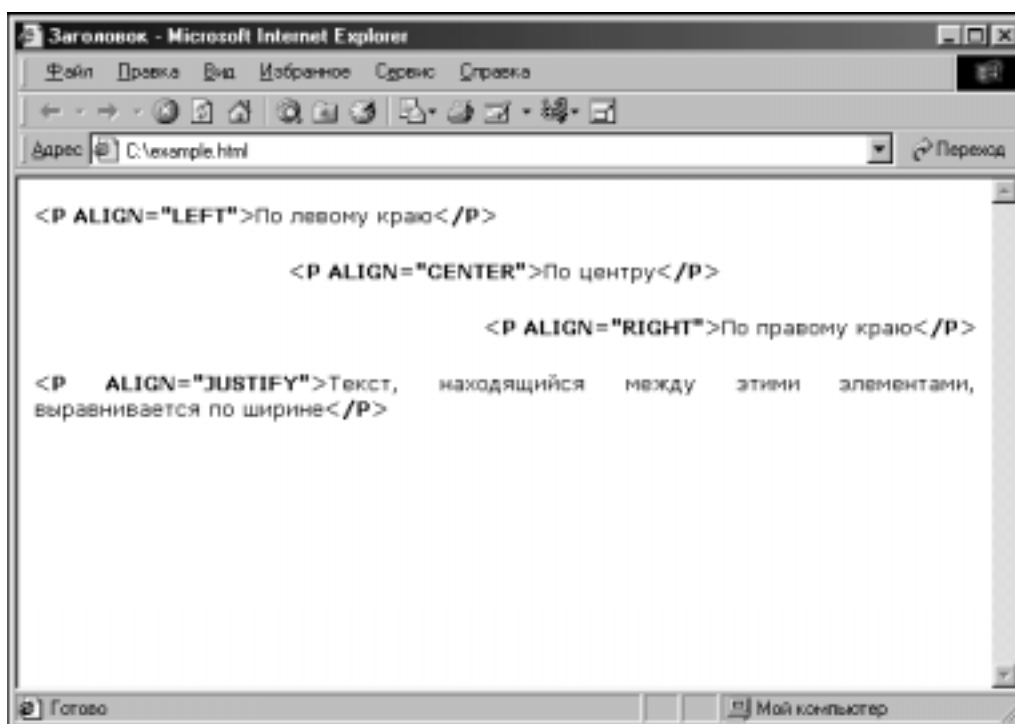


Рис. 1.12. Пример использования различных вариантов выравнивания текста

Для придания шрифту, которым выводится текст, определенного начертания (полужирный, курсив, подчеркнутый и т.д.), поместите этот текст между парными дескрипторами, перечисленными ниже. Результат их применения представлен на рис. 1.13.

- — полужирное начертание.
- <I> </I> — курсивное начертание.
- <U> </U> — подчеркнутый текст.
- <STRIKE> </STRIKE> — перечеркнутый текст.
- — верхний индекс.
- — нижний индекс.

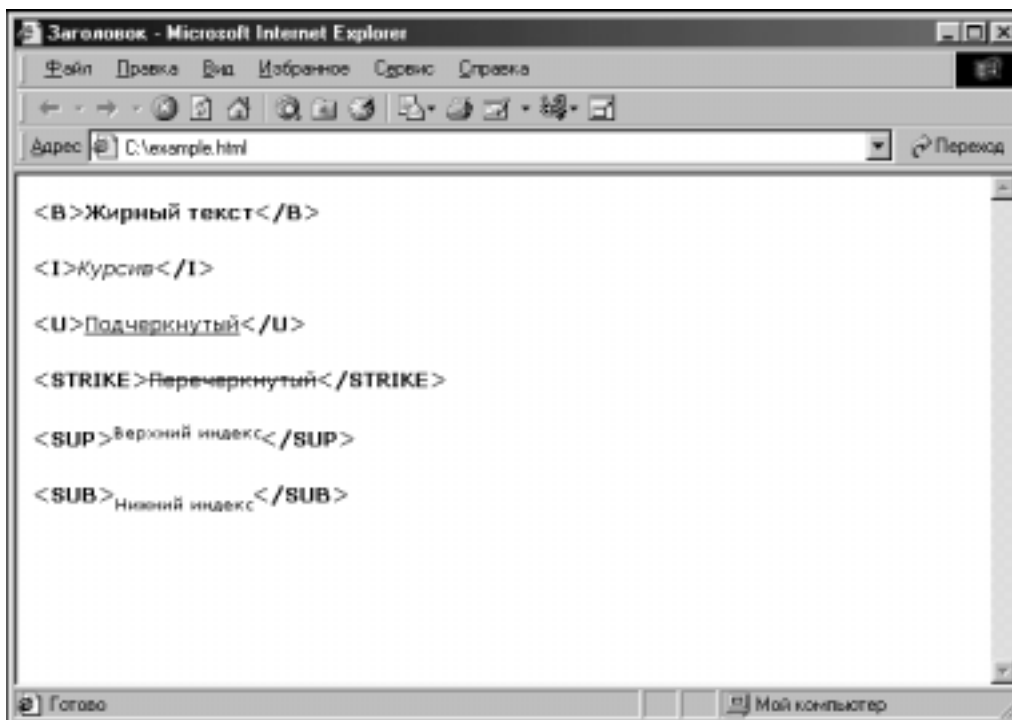


Рис. 1.13. Пример придания шрифту текста различных вариантов начертания

К одному и тому же фрагменту текста может применяться сразу несколько подобных дескрипторов, как показано ниже.

```
<tt><b><i>Текст</i></b></tt>
```

Некоторые дескрипторы могут или должны применяться с определенными параметрами (или атрибутами, называйте их, как хотите), которые указываются в открывающем элементе дескриптора (в одном дескрипторе одновременно можно указывать несколько различных параметров). Например, открывающий дескриптор (закрывающий дескриптор обязателен) может иметь несколько параметров, описываемых ниже.

- SIZE — задает размер шрифта (по умолчанию размер шрифта равен 3). Поместив текст между дескрипторами , где *n* — цифровое значение в диапазоне от 1 до 6, можно придать используемому шрифту указанный размер (рис. 1.14).
- FACE — задает стандартное имя используемого шрифта. Здесь очень **важно** всегда использовать шрифты, которые с большой вероятностью могут быть установлены на компьютере пользователя. В противном случае браузер просто подставит шрифт, определенный по умолчанию (обычно Times New Roman), и все ваши усилия окажутся напрасными. К стандартным шрифтам можно отнести шрифты, поставляемые с Windows, пакетами Microsoft Plus или Microsoft Office.

Допустим, вы создаете HTML-документ, используя шрифт Comic Sans MS, а на компьютере пользователя этот шрифт был удален (случайно или сознательно, это для нас неважно). Как лучше выйти из подобной ситуации, ведь браузер, не найдя этот шрифт на компьютере пользователя, отобразит весь имеющийся в документе текст шрифтом Times New Roman. В результате все ваши оформительские усилия будут по-

трачены впустую. Тут поможет указание так называемого семейства шрифтов, при этом имя каждого отдельного шрифта пишется через запятую. В документе, показанном на рис. 1.15, в последней строке используется шрифтовое семейство: если у пользователя на компьютере не окажется шрифта Comіc Sans MS, браузер подставит следующий шрифт из указанного списка — Tahoma.

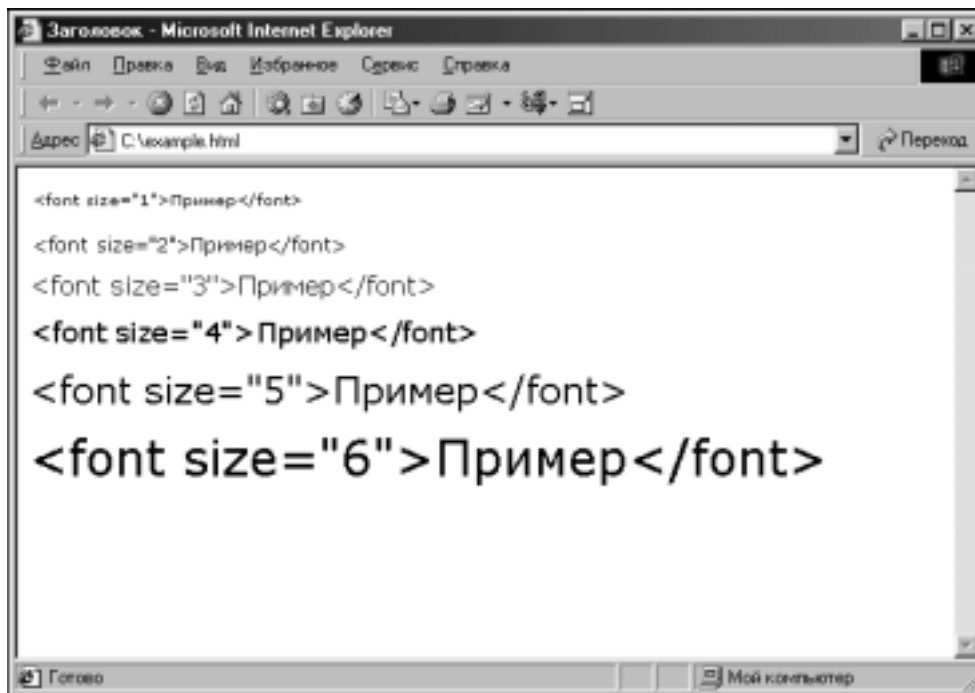


Рис. 1.14. Придание шрифту текста определенного размера

Параметр `COLOR` предназначен для указания цвета, которым выводится текст (по умолчанию используется черный цвет — код #000000). Цвет текста может определяться как его названием, так и шестнадцатеричным кодом. Примеры различных способов определения параметра `COLOR` приведены ниже.

```
<font COLOR="red">Красный</font>  
<font COLOR="#FF0000">Красный</font>
```

Рассмотренные выше параметры дескриптора `` могут использоваться все вместе или в любых сочетаниях, в зависимости от того, как требуется отформатировать создаваемый документ. Ниже приведен пример, где задается использование для отображения текста шрифта Verdana синего цвета и относительного размера 2.

```
<font face="Verdana" color="blue"  
size="2">Пример</font>
```

С помощью параметра `STYLE` парного дескриптора `` (закрывающий дескриптор `` здесь обязателен) можно задавать выделение текста любым цветом, как показано на рис. 1.16.

Такое форматирование создает так называемые стили (CSS — набор правил оформления и форматирования, который может быть применен к различным элементам страницы). Более подробно о них речь пойдет в последующих главах книги.

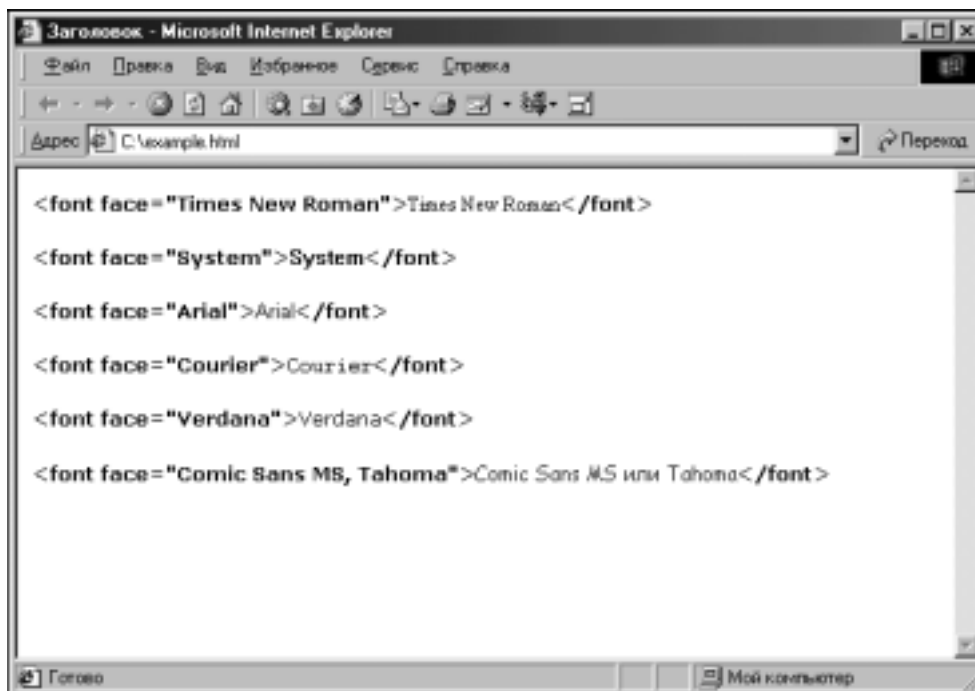


Рис. 1.15. Пример использования в HTML-документе различных шрифтов

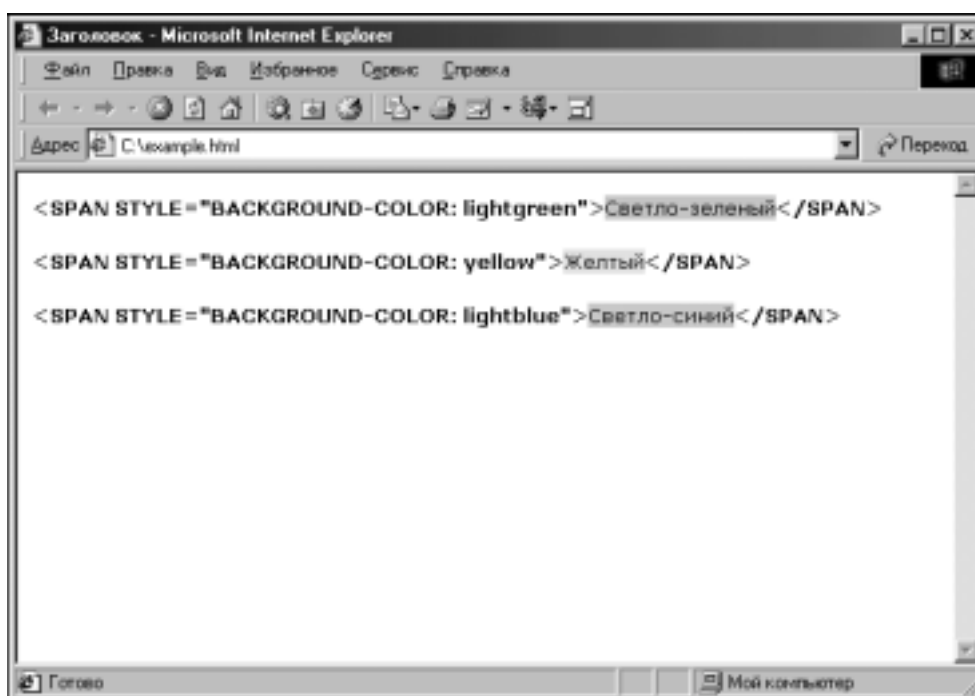


Рис. 1.16. Пример использования различных вариантов выделения текста

Если желательно сохранить текст именно в таком виде, как он был набран, необходимо поместить его между парными дескрипторами `<pre>` `</pre>`. Текст между этими дескрипторами будет выводиться с точностью до пробелов именно так, как он был набран в приложении Блокнот. Данный дескриптор может быть полезен, допустим, для форматирования отступов в стихотворениях.

```
<pre>текст      (множество пробелов)      текст</pre>
```

Списки

Если вам нужно разместить на странице нумерованный список (например, названий каких-то товаров), воспользуйтесь дескриптором ``. Обработывая его, браузер будет автоматически управлять нумерацией списка, — естественно, по заранее составленному вами плану. Начинается нумерованный список с дескриптора `` и завершается дескриптором ``. Исходный HTML-код описания списка и его отображение в браузере представлены ниже (рис. 1.17).

```
<OL>
<LI>Один
<LI>Два
<LI>Три
<LI>Четыре
<LI>Пять
</OL>
```

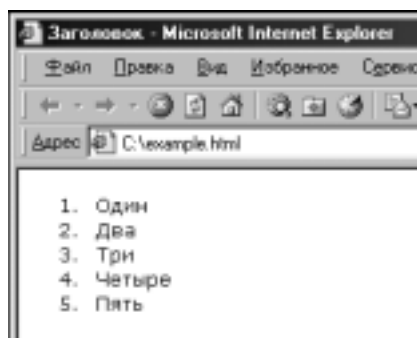


Рис. 1.17. Пример отображения нумерованного списка

Дескриптор `` может содержать параметр `TYPE`, которым задается тип используемой нумерации.

- A — большие латинские буквы.
- a — маленькие латинские буквы.
- I — большие римские цифры.
- i — маленькие римские цифры.
- 1 — арабские цифры.

Так, для примера, приведенного на рис. 1.17, параметр `TYPE` будет иметь значение `TYPE=1`. Также можно задать начальное значение, с которого будет начинаться отсчет. Для этого предназначен параметр `START`.

Приведенный ниже пример описывает список, пронумерованный большими латинскими буквами и начинающийся с цифры 8 (рис. 1.18).

```
<OL TYPE="I" START="8">
<LI>Восемь
<LI>Девять
<LI>Десять
<LI>Одиннадцать
<LI>Двенадцать
</OL>
```

Для нумерованных списков (такие тоже бывают и называются *маркированными*) браузер для пометки элемента списка обычно использует специальные маркеры — это или небольшой кружочек (заштрихованный или не заштрихованный), или же заштрихованный квадратик.

Описание маркированного списка начинается открывающим дескриптором и завершается его парным дескриптором . Каждый элемент списка в этом случае также начинается с дескриптора . Пример описания нумерованного (маркированного) списка и его представление в браузере (рис. 1.19) приведены ниже.

```
<UL>
<LI>Один
<LI>Два
<LI>Три
<LI>Четыре
<LI>Пять
</UL>
```

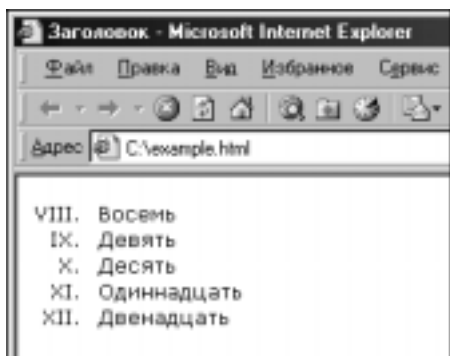


Рис. 1.18. Пример списка с нумерацией римскими цифрами, начиная со значения VIII

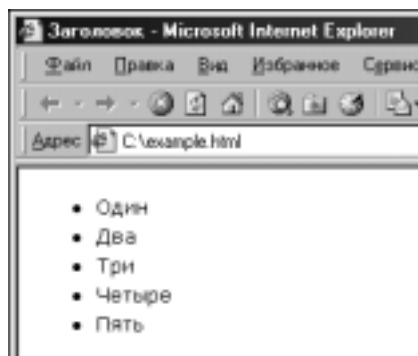


Рис. 1.19. Пример маркированного списка

Дескриптор может включать параметр TYPE со значениями disc, circle или square, определяющими вид используемого маркера. По умолчанию маркер ставится в виде заштрихованного кружочка, т.е. используется значение disc. Если один список вкладывается в другой, то маркеры вложенного списка будут незаштрихованными. Чтобы избежать этих выполняемых по умолчанию действий, явно указывайте требуемое значение параметра TYPE. Примеры определения типа маркеров и их внешний вид (рис. 1.20) показаны ниже.

```
<UL TYPE="disc"><LI>disc</LI>
<UL TYPE="circle"><LI>circle</LI>
<UL TYPE="square"><LI>square</LI>
```

В языке HTML существует еще один вариант списков — списки определений. Способ их описания с помощью дескрипторов <DL> </DL> и <DD>, а также вид этих списков в окне браузера (рис. 1.21) приведены ниже.

```
<DL>
<DT>Термин
<DD>Определение термина 1.
<DD>Определение термина 2.
<DD>Определение термина 3.
<DD>Определение термина 4.
</DL>
```

В дескрипторе <DL> может присутствовать параметр COMPACT, позволяющий разместить пункты списка более компактно. Данному параметру не требуется присваивать каких-либо значений.

```
<DL COMPACT>
```

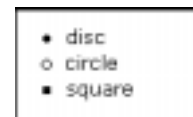


Рис. 1.20. Вид доступных маркеров нумерованных списков

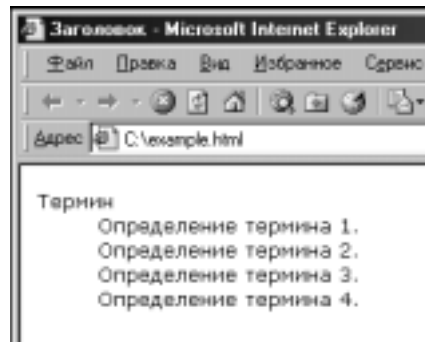


Рис. 1.21. Пример списка определений

Резюме

В этой вводной главе вы узнали, какие инструменты понадобятся начинающему Web-мастеру и зачем. Далее был приведен пример создания простенького, но зато самого первого собственного HTML-документа, и вы впервые увидели в окне браузера результаты проделанной работы. Далее обсуждались вопросы выбора и описания цвета фона Web-страницы, а также методы форматирования текста в HTML-документах. В завершение были рассмотрены способы создания HTML-списков и приведены примеры их различных вариантов.