

В. Байков



ИНТЕРНЕТ

Поиск информации
Продвижение сайтов



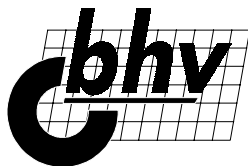
WWW PRICE RU

ВСЕГДА ПОД РУКОЙ!

Владимир Байков

ИНТЕРНЕТ:

**поиск информации
и продвижение сайтов**



Санкт-Петербург

Дюссельдорф ♦ Киев ♦ Москва ♦ Санкт-Петербург

В книге рассматриваются две взаимосвязанные технологии — поиск информации и продвижение (раскрутка) сайтов. Изложение указанных вопросов основывается на достаточно детальном описании структуры документов Интернета на базе языка разметки гипертекстов, а также основных принципов работы поисковых серверов, без знания которых невозможен ни эффективный поиск информации в Интернете, ни эффективное рекламирование сайтов. Автор также детально анализирует некоторые распространенные ошибочные стратегии рекламирования сайтов.

Значительное внимание в книге уделено мало освещенным в современной отечественной и переводной литературе вопросам — поиску и рекламированию документов Интернета через сервисы электронной почты, что особенно актуально в связи с реальным переходом на повременную форму оплаты телефонных услуг.

Для широкого круга читателей

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Наталья Таркова</i>
Редакторы:	<i>Татьяна Кручинина, Владимир Овчинников</i>
Компьютерная верстка:	<i>Натальи Смирновой, Ольги Сергиенко</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Ангелины Лужиной</i>
Зав. производством	<i>Николай Тверских</i>

Байков В. Д.

Интернет: поиск информации и продвижение сайтов. — СПб.: БХВ — Санкт-Петербург, 2000. — 288 с.: ил.

ISBN 5-8206-0095-9

© В. Д. Байков, 2000

© Оформление, издательство "БХВ — Санкт-Петербург", 2000

Лицензия ЛР № 065953 от 15.06.98. Подписано в печать 22.06.00.

Формат 60×90^{1/16}. Печать офсетная. Усл. печ. л. 18.

Тираж 3000 экз. Заказ

"БХВ — Санкт-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН.
199034, Санкт-Петербург, 9-я линия, 12.

Содержание

Введение	5
Глава 1. Структура Web-документов	7
1.1. Общая структура Web-документа	11
Обязательные метки	13
Блочные элементы разметки	13
Горизонтальные разделительные линии	17
1.2. Шрифтовые выделения и списки.....	17
Списки	19
1.3. Организация таблиц	26
Основные элементы таблицы	26
1.4. Гипертекст, якоря и гиперссылки.....	31
1.5. Цветовая гамма и графические изображения в Web-документах.....	37
Цветовая гамма	37
Встраивание изображений	42
Элементы анимации текста	46
1.6. Организация фреймов.....	47
1.7. Навигационные карты.....	51
1.8. Формы	55
Пример простейшей формы	57
Как форма собирает данные	57
Основные типы элементов.....	58
1.9. Использование метатегов.....	65
Пример создания слайд-презентации	71
Глава 2. Эффективные методы поиска информации в Интернете	74
2.1. Виды информационных ресурсов и инструментов поиска	74
2.2. Тематические каталоги.....	83
Англоязычные каталоги	83
Русскоязычные каталоги	94
2.3. Поиск по индексным базам данных	101
Англоязычные и мультязычные поисковые системы.....	101
Русскоязычные поисковые системы	108
2.4. Специализированные поисковые службы	112
2.5. Метапоиск и его возможности	124
2.6. Организация усложненного поиска	132
2.7. Особенности поиска документов на русском языке	137
2.8. Поиск специальных элементов	140
2.9. Поиск в группах телеконференций.....	144
2.10. Можно ли найти в Интернете новости СМИ?	159

2.11. Поиск программного обеспечения.....	165
2.12. Поиск информации в Интернете через сервисы электронной почты	173
Доступ к FTP и Archie через электронную почту.....	175
Поиск в WWW через электронную почту.....	182
Поиск в группах телеконференций.....	192
Глава 3. Методы продвижения сайтов в Интернете.....	197
3.1. Виды рекламы сайтов в Интернете.....	197
3.2. Основные принципы работы поисковых серверов.....	201
AltaVista.....	206
InfoSeek.....	208
Lycos.....	209
WebCrawler.....	210
HotBot.....	211
Northern Light.....	212
Google.....	212
Апорт.....	214
3.3. Выбор метатегов — основа саморекламы сайтов.....	215
3.4. Техника поиска бесплатных списков регистрации.....	233
3.5. Реклама в телеконференциях и списках рассылки.....	240
3.6. Популярные "трюки" раскрутки сайтов и их разоблачение.....	253
3.7. Как правильно раскрутить сайт?.....	262

Мы не в силах повысить качество

Вашего Интернет-соединения...

Но мы можем повысить эффективность его работы



ReGet оптимизирован для работы на российских телефонных линиях. Ему не страшны обрывы связи, плохие каналы, загруженные серверы. **ReGet** – надежный попутчик в путешествиях по Сети.

Скачай себе бесплатную версию

www.reget.com

Введение

Ни с чем не сравнимая скорость развития Интернета как с точки зрения количества пользователей и компьютеров, входящих в Сеть, так и с точки зрения роста числа документов Сети ставит ряд задач, связанных с эффективным поиском информации и эффективным рекламированием (продвижением, раскруткой) вебсайтов в Интернете. Другими словами, надо не потерять в Сети нужную информацию и не потеряться там самим с созданными вами Web-сайтами, т. е., как говорится, и на других посмотреть, но и себя суметь показать. А посмотреть в Интернете есть на что — около 500 миллионов документов находится там сегодня, и число их ежегодно удваивается.

Как же эффективно искать информацию и как помочь в ее поиске тем, кто в ней может быть заинтересован? Конечно, определяющую роль тут играют различные поисковые серверы. Число их постоянно растет, становятся более разнообразными и виды их услуг. Простейшие виды поиска информации и продвижения сайтов, не требующие ни особых навыков, ни специальных знаний, ушли в безвозвратное прошлое. Такие приемы могли эффективно работать только тогда, когда число документов Сети составляло десятки или сотни тысяч. Тогда даже и в регистрации сайтов особой необходимости не было: поисковые серверы сами "прочесывали" весь Интернет и, обнаружив новый сайт, вносили его в свои базы данных. Сейчас дело совсем иное: ни один поисковый сервер не может гарантировать, что в нем представлены все или, по крайней мере, большая часть документов Интернета.

Если попытаться отыскать сейчас в Интернете нужную информацию по одному-двум словам поиска (ключевым словам), то в ответ вы скорее всего получите миллионы если не десятки миллионов адресов документов. Значит, нужны более тонкие механизмы поиска, учитывающие структуру самих документов и их особенности. Это же нужно учитывать и при продвижении Web-сайтов в Интернете.

Следовательно, без знания принципов построения самих документов Сети и без изучения особенностей работы поисковых серверов ни эффективного поиска информации, ни эффективного продвижения сайтов реализовать нельзя. Поэтому структура книги и следует логике изучения материала. Вначале излагается язык разметки гипертекстов — HTML, на базе которого строится подавляющее большинство документов Интернета. Затем последовательно описывается техника поиска различного вида информации. После чего излагаются виды продвижения сайтов в Интернете.

Глава 1



Структура Web-документов

Разнообразие внешнего вида Web-документов наверное сравнимо только с разнообразием обложек журналов, газет и рекламы: здесь различные шрифты, иллюстрации, анимационные фигуры и пр. Кроме того, можно увидеть и специфические только для документов Интернета разнообразные интерактивные формы, предназначенные для заполнения пользователем и отправки их обратно на сервер. Образцы оформления Web-документов представлены на рис. 1.1—1.5.

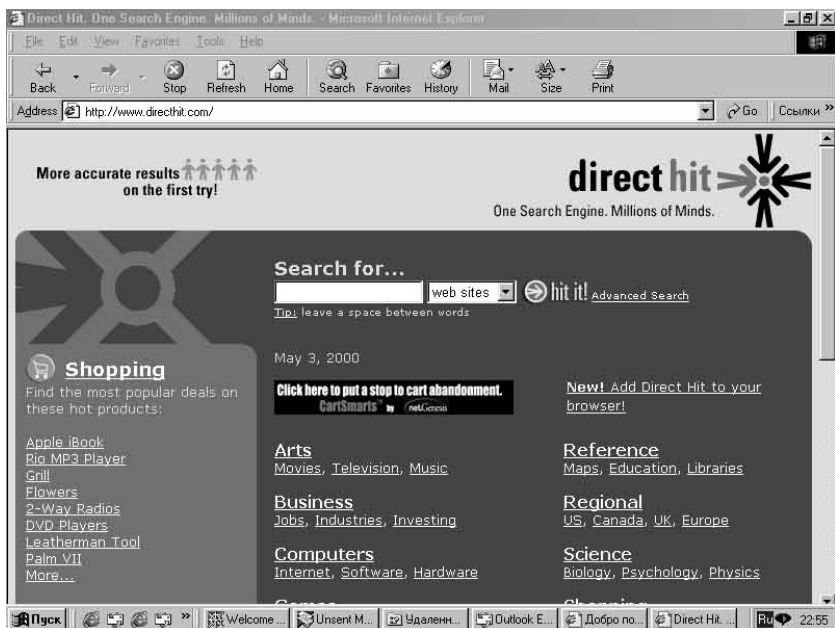


Рис. 1.1. Пример оформления каталога DirectHit

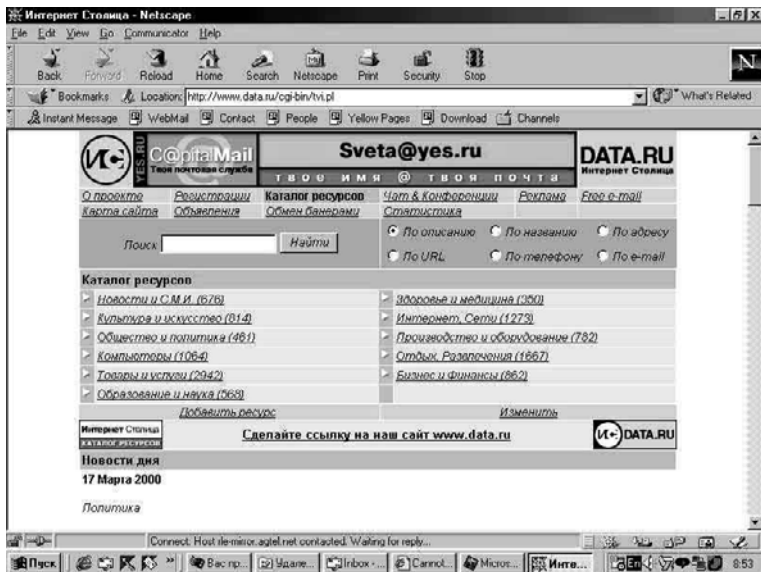


Рис. 1.2. Пример оформления русскоязычного поискового сервера

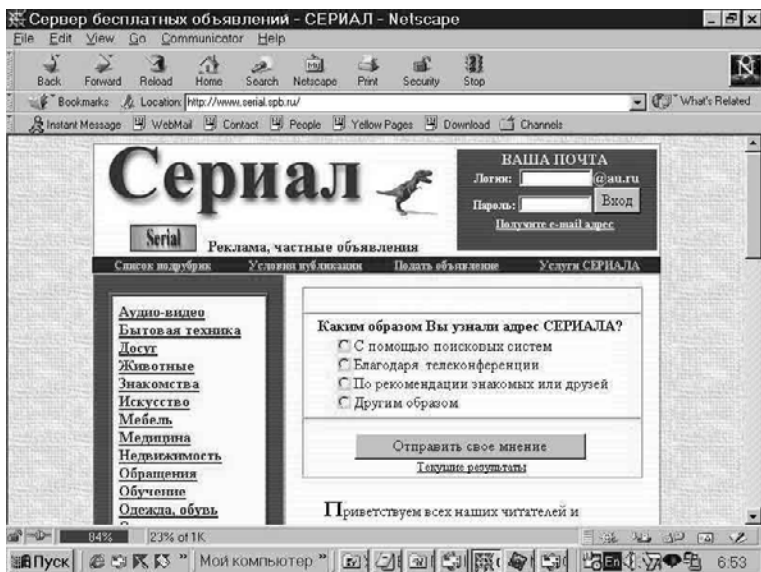


Рис. 1.3. Бизнес-сайт, предоставляющий бесплатные услуги в русскоязычной части Сети

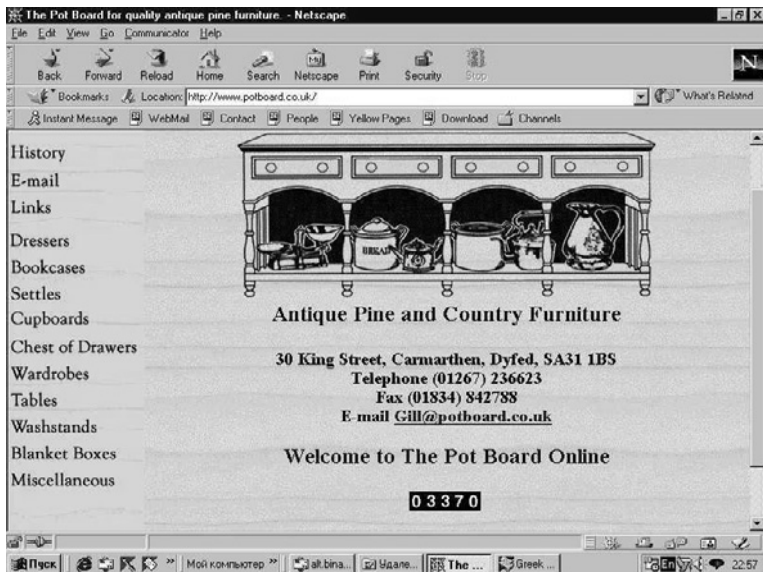


Рис. 1.4. Выставка-продажа антиквариата

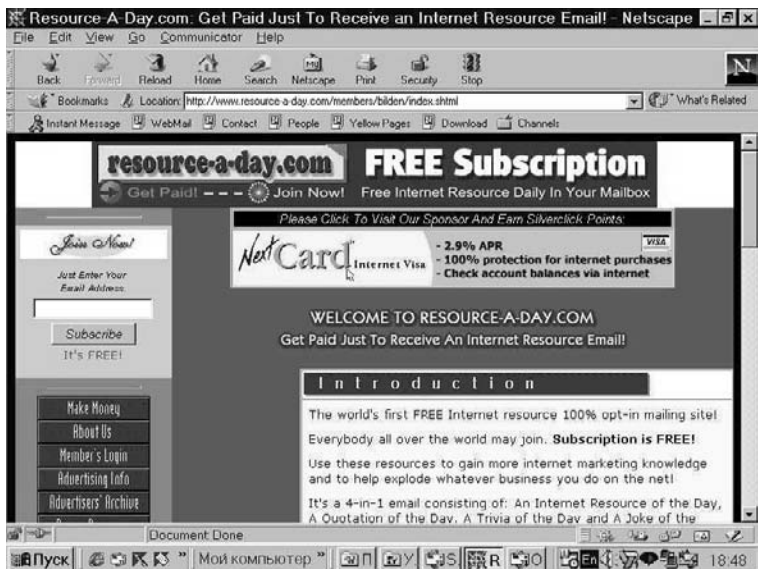


Рис. 1.5. Пример сайта, предлагающего бесплатные услуги по размещению рекламы в Интернете

Самое удивительное, что все эти документы строятся на базе единого, и притом довольно простого языка HTML — языка разметки гипертекстов. "Изнанка" всех документов выглядит примерно так, как показано на рис. 1.6.

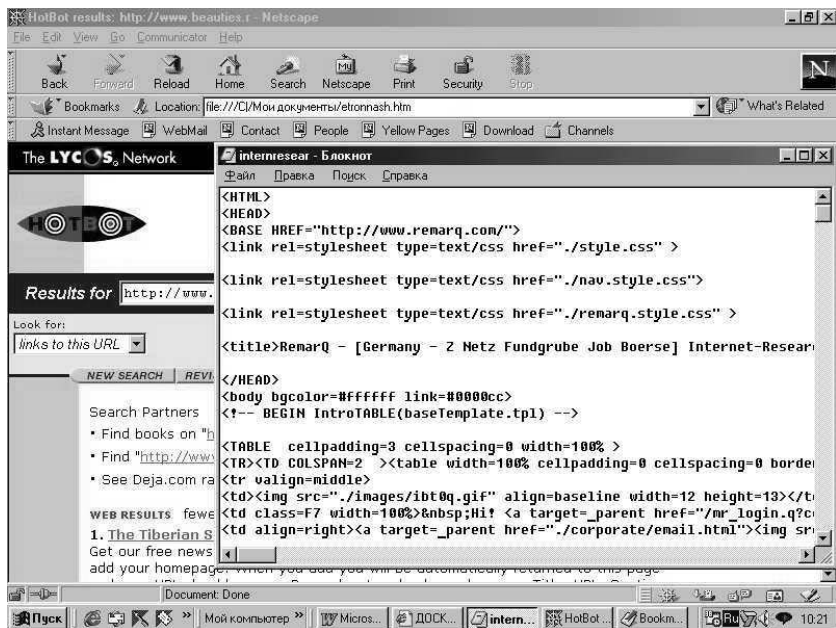


Рис. 1.6. Внешний вид Web-документа и его "изнанка"

На заднем плане показана лицевая сторона документа, т. е. то, что видит пользователь, а на переднем плане в черно-белом изображении — внутренняя структура документа, его рабочее представление. Вероятно, так же уныло выглядят в виде временной диаграммы импульсов сигналы телевидения, которые на экране преобразуются в яркую, подвижную, да еще и звучащую картину. Для нас же самое главное с точки зрения основных целей книги не столько показать, как эти однообразные значки преобразуются в изображения, анимацию, звук, а как они влияют на методы и время поиска Web-документов в Интернете, а также как, зная их структуру, научиться эффективно продвигать и раскручивать свой Web-сайт, создание которого стоит очень и очень недешево.

1.1. Общая структура Web-документа

Сам термин *HTML* (HyperText Markup Language) означает *язык разметки гипертекстов*. Это действительно в первую очередь язык разметки, а не язык программирования, поэтому все, кто имеет отношение к редакционно-издательской деятельности, осваивают его сравнительно легко и быстро.

Язык HTML будет интересовать нас не сам по себе, а, главным образом, с точки зрения овладения эффективными технологиями поиска информации и рекламирования Web-сайтов в Интернете. Для этих целей доскональное знание структуры Web-документации, которая строится на базе языка HTML, просто необходимо.

Web-документ — это текстовый файл с расширением .htm или .html. Пример простого Web-документа:

```
<html>
  <head>
    <title>
      </title>
  </head>
  <body>
    <h1>
```

Первое знакомство

```
</h1>
  <p>
```

Пример Web-документа.

```
<p>
```

```
</body>
</html>
```

Для удобства чтения на этапе составления и отладки вводятся переводы строки, которые затем с целью экономии, а также сохранения авторской тайны, т. е. know-how, могут быть устранены. Сами же браузеры просто игнорируют символы конца строки и множественные пробелы в HTML-файлах. Поэтому данный пример вполне мог бы выглядеть вот так:

```
<html><head><title> Пример1</title></head>  
<body> <n1>Простейший пример</n1>  
Пример Web-документа.<br>  
Этот *.html-файл может быть одновременно открыт в Notepad, в  
Explorer, и в Netscape.<br>  
</body> </html>
```

Это будет выглядеть на экране браузера так, как показано на рис. 1.7.

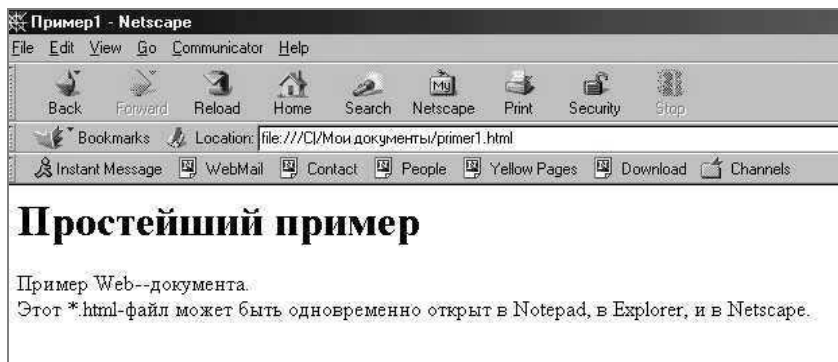


Рис. 1.7. Простейший Web-документ

Как видно на рис. 1.7, вся информация о форматировании документа сосредоточена в его фрагментах, заключенных между открывающей и закрывающей угловыми скобками, т. е. между знаками "<" и ">". Такой фрагмент (например, <html>) называется *меткой* или *тегом*. Большинство HTML-меток — парные, т. е. на каждую открывающую метку вида <...> есть закрывающая метка вида </...> с тем же именем, но с добавлением косой черты "/".

Метки можно вводить как прописными, так и строчными буквами. Например, метки <body>, <BODY> и <body> будут восприняты любым браузером одинаково. Многие метки, помимо имени, могут содержать атрибуты, т. е. элементы, дающие дополнительную информацию о том, как браузер должен обработать текущую метку. По мере изложения материала будем вводить различные атрибуты.

Обязательные метки

Метки `<html> ... </html>`

Метка `<html>` должна открывать HTML-документ, а метка `</html>` — завершать его.

Метки `<head> ... </head>`

Эта пара меток указывает на начало и конец заголовка документа. Помимо наименования документа (см. описание метки `<title>` ниже), в этот раздел может включаться служебная информация, о которой мы обязательно поговорим чуть позже.

Метки `<title> ... </title>`

Все, что находится между метками `<title>` и `</title>`, толкуется браузером как название документа. Браузер Netscape Communicator, например, показывает название текущего документа в заголовке окна и печатает его в левом верхнем углу каждой страницы при выводе на принтер. Рекомендуется давать документу название, содержащее от 60 до 100 символов.

Метки `<body> ... </body>`

Эта пара меток указывает на начало и конец тела HTML-документа, внутри которых и работают все стандартные теги.

Блочные элементы разметки

Метки заголовков `<H1> ... </H1> — <H6> ... </H6>`

Метки вида `<hi>` (где *i* — цифра от 1 до 6) описывают относительные размеры заголовков шести различных уровней. Заголовок первого уровня — самый крупный, шестого уровня, естественно, — самый мелкий (рис. 1.8).

Метки абзаца `<P> ... </P>`

Такая пара меток описывает абзац. Все, что заключено между `<P>` и `</P>`, воспринимается как один абзац. Закрывающий тег — `</P>` необязателен, поскольку сама программа может поставить его в нужное место.

Метка `<P>` может содержать дополнительный атрибут `ALIGN`, принимающий одно из значений: `left`, `center`, `right`. Например:

<P ALIGN=CENTER> – выравнивание параграфа по центру
ИЛИ

<P ALIGN=RIGHT> – абзац с выравниванием по правому краю

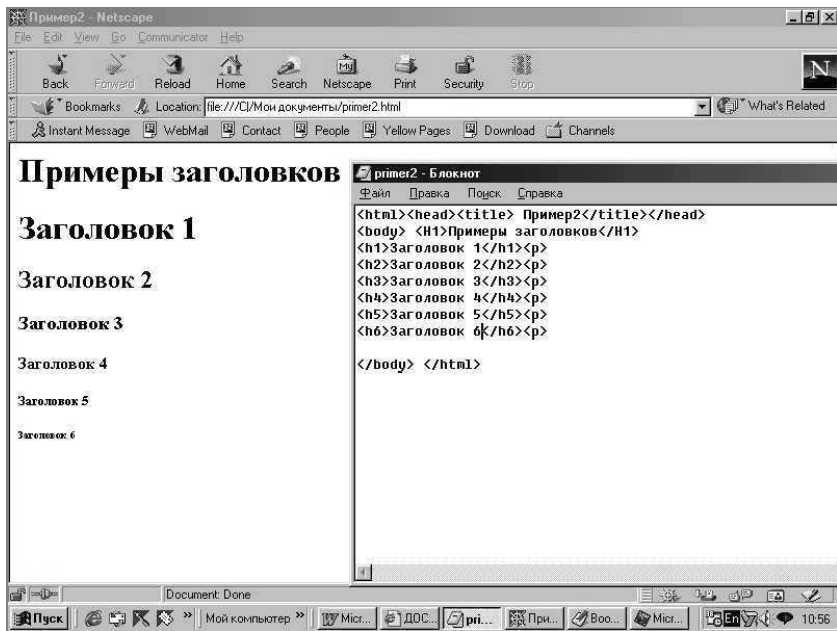


Рис. 1.8. Варианты заголовков

Точно такие же значения может принимать атрибут <ALIGN> при тегах <h1>.

Приведем соответствующий пример:

```

<html>
<head>
<title> Пример 3 </title>
</head>
<body>
<h1 ALIGN=right>Прижатый вправо заголовок</h1>
<h2 ALIGN=center>Заголовок по центру</h2>
<P ALIGN=CENTER> <b>Выравнивание по центру</b>
  
```

Индивидуальность передоверяет свою судьбу мифологическому институту — партии, правительству, Старшему Брату. И за это она освобождается от мучений, которые приносит личная ответственность. В мире, где правит миф, легче быть счастливым" (конец цитаты).

</p>

<P ALIGN=left> Выравнивание по левому краю

Индивидуальность передоверяет свою судьбу мифологическому институту — партии, правительству, Старшему Брату. И за это она освобождается от мучений, которые приносит личная ответственность. В мире, где правит миф, легче быть счастливым" (конец цитаты). Здесь подчеркнута важная особенность мифологического поведения — отказ от осмысленного решения в пользу эмоционально — образных побуждений. Миф — принадлежит к классу моделей действительности

</p>

<P ALIGN=RIGHT> Выравнивание по правому краю

Мифодизайнеры неточны, они называют мифом легенду. Кроме образности, эмоциональной окрашенности и эксплуатации скрытых, неявных ожиданий настоящий миф претендует на широкий охват реальности. Петр Вайль и Александр Генис писали об...

</p>

</body>

</html>

Мы увидим на экране варианты расположения абзацев (рис. 1.9).

Метка конца строки

В том случае, когда нужно не организовывать новый абзац, а просто перейти на новую строку, используется тег
. Примером этого является организация формата стихов или песен:

```
<I><B>Strangers In The Night</B></I><P>
Strangers in the night exchanging glances<BR>
Wond'ring in the night what were the chances <BR>
We'd be sharing love<BR>
Before the night was through.<BR>
Something in your eyes was so inviting,<BR>
Something in you smile was so exciting,<BR>
Something in my heart,<BR>
Told me I must have you...<BR>
```

Результат показан на рис. 1.10.

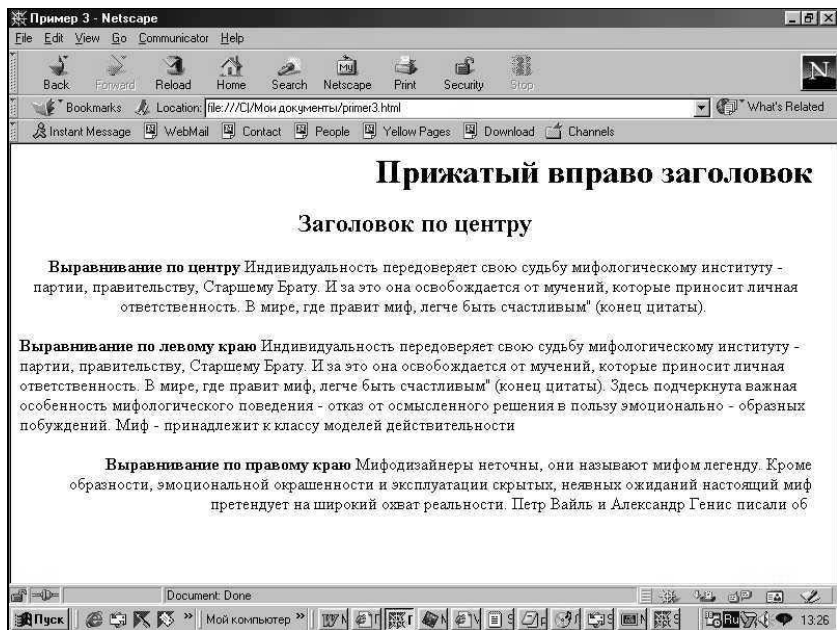


Рис. 1.9. Варианты расположения абзацев

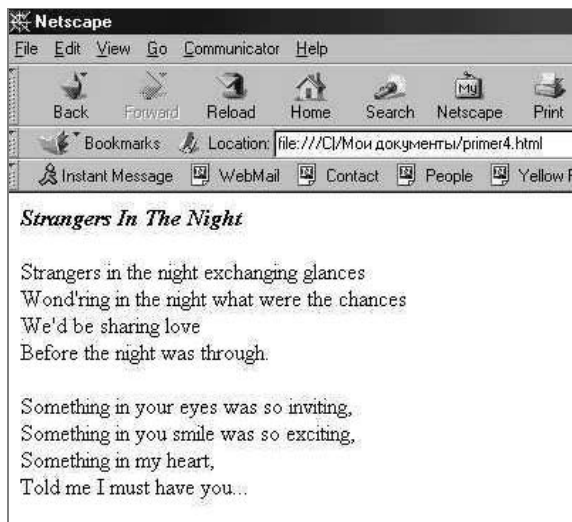


Рис. 1.10. Использование меток конца строки

Горизонтальные разделительные линии

Метка `<HR>` описывает вот такую горизонтальную линию:

Метка `<HR>` может также дополнительно включать атрибуты `SIZE` (определяет толщину линии в пикселах) и/или `WIDTH` (определяет размер линии в процентах от ширины экрана). Для браузера Microsoft Internet Explorer работает также атрибут `COLOR`. Ниже приведен ряд примеров построения горизонтальных линий.

```
<H5> Коллекция горизонтальных линий</H5>
```

```
<HR SIZE=2 WIDTH=100% color=green>
```

```
<HR SIZE=4 WIDTH=50% color=red>
```

```
<HR SIZE=8 WIDTH=25% color=green>
```

```
<HR SIZE=16 WIDTH=12% color=grey>
```

На экране увидим разноцветные линии, отображенные в черно-белом варианте на рис. 1.11.

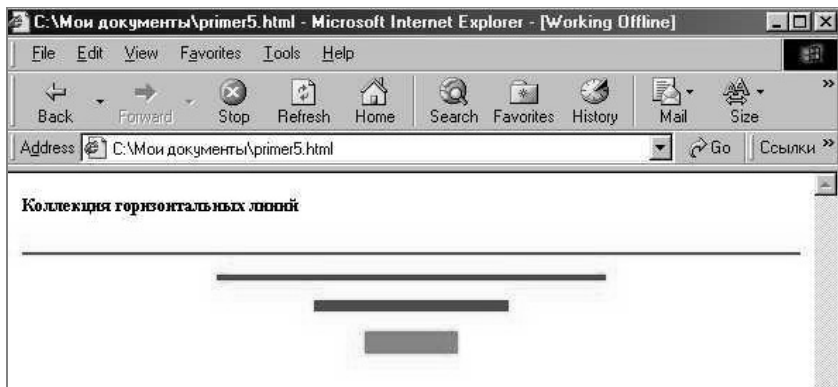


Рис. 1.11. Виды задания параметров разделительных линий

1.2. Шрифтовые выделения и списки

Остановимся на так называемом *физическом стиле* шрифтового выделения. Под физическим стилем принято понимать прямое

указание браузеру на модификацию текущего шрифта. Например, все, что находится между метками `` и ``, будет выделено жирным шрифтом, а между метками `<I>` и `</I>` — курсивом. Текст между метками `<U>` и `</U>` будет подчеркнутым. Кроме того, можно включать в текст фрагменты с фиксированной шириной символа `<TT>` (имитация пишущей машинки) `</TT>`. Существует еще ряд выделений: шрифт, более крупный, чем основной:

```
<big>большой</big>
```

или более мелкий

```
<small>маленький</small>
```

Можно также применять надстрочные `^{X...}` или подстрочные `₁₂₃₄` надписи. Приведем сводный пример:

```
<b>Полужирный шрифт</b><br>
```

```
<i>Курсив</i><br>
```

```
<u>Подчеркнутый шрифт</u><br>
```

```
<tt>Шрифт пишущей машинки</tt><br>
```

```
<big>Крупный шрифт</big><br>
```

```
<small>Мелкий шрифт</small><br>
```

```
x<sup>342</sup> так пишутся показатели степени<br>
```

```
a<sub>i</sub> , а так — индексы<br>
```

Все это на экране выглядит так, как показано на рис. 1.12.

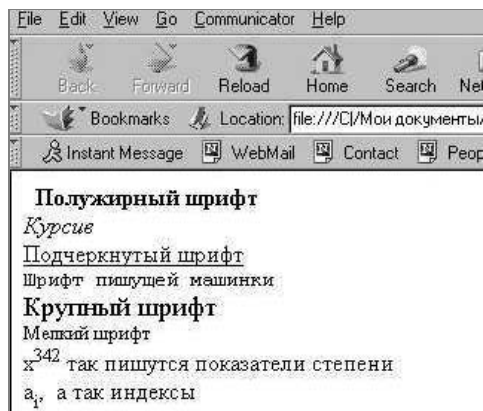


Рис. 1.12. Виды шрифтов

Существуют также способы цветового выделения шрифта (рис. 1.13), но о них мы поговорим в параграфе, целиком посвященном цвету.

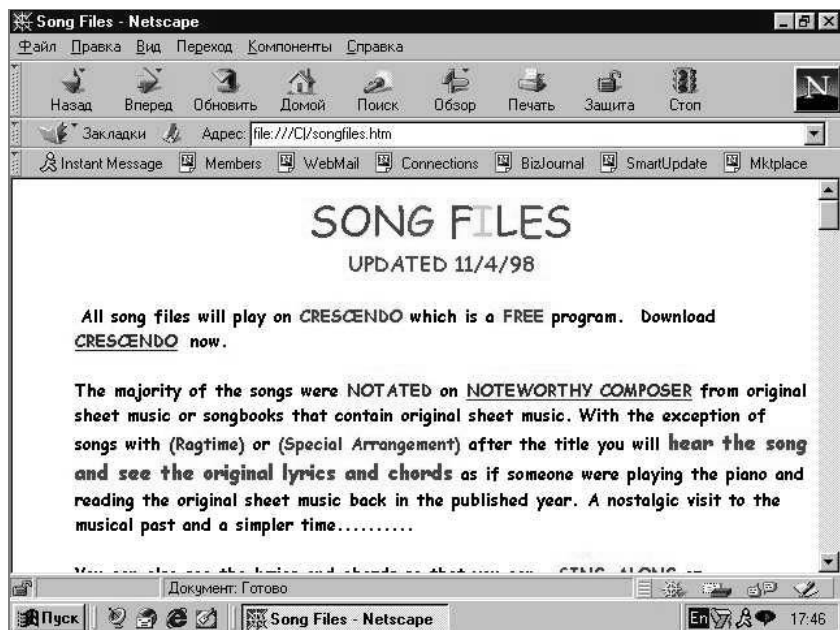


Рис. 1.13. Выделение шрифтов: видом и цветом

Списки

Язык HTML позволяет определять внешний вид отдельных абзацев текста. Абзацы можно организовывать в списки, выводить их на экран в отформатированном виде. Язык HTML поддерживает несколько видов списков.

Ненумерованные списки ` ... `

Текст, расположенный между метками `` и ``, воспринимается как ненумерованный (маркированный) список. Каждый новый элемент списка следует начинать с метки ``. Например, чтобы создать список, представленный на рис. 1.14, необ-

ХОДИМО записать так: `Первый студент Второй студент Третий студент `.

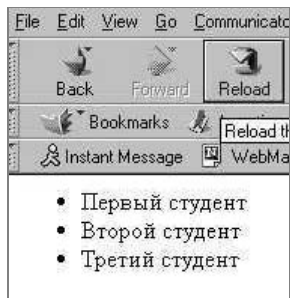


Рис. 1.14. Вариант списка

Разделители

Для отделения элементов маркированного списка друг от друга предназначены следующие разделители:

- `<li type=disc>` — элементы отделяются заштрихованным диском;
- `<li type=circle>` — элементы отделяются незаштрихованным кругом;
- `<li type=square>` — элементы отделяются заштрихованным квадратом.

Возможно использование различных разделителей для каждого элемента списка. Например, таких:

- ``;
- `` первый элемент списка;
- `` второй элемент списка;
- `<li type=square>` третий элемент списка;
- `<li type=disc>` четвертый элемент списка;
- `<li type=circle>` пятый элемент списка ``.

Неупорядоченный список показан на рис. 1.15.

Упорядоченные списки `...`

Для составления, например, упорядоченного (нумерованного) списка, представленного на рис. 1.16, необходимо составить такой Web-документ:

```
<OL>  
<li> Первая студентка  
<li> Вторая студентка  
<li> Третья студентка  
</OL>
```

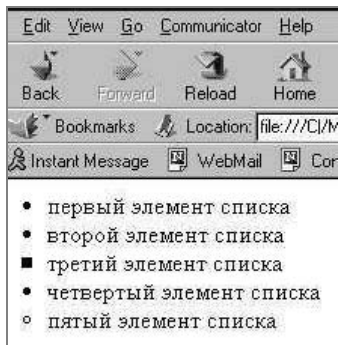


Рис. 1.15. Неупорядоченный список

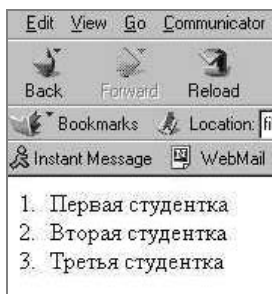


Рис. 1.16. Нумерованный список

С помощью атрибута `type` также можно задать различные виды нумерации:

- `<li type=1>` — арабские числа;
- `<li type=a>` — строчные буквы;
- `<li type=A>` — заглавные буквы;
- `<li type=i>` — малые римские цифры;
- `<li type=I>` — большие римские цифры.

В случае нумерованных списков, так же, как и для нумерованных, можно комбинировать для каждой строчки типы разде-

лителей, используя общую сквозную нумерацию. Например, начать с арабских чисел, затем перейти на римскую нумерацию, а после использовать латинские цифры. Приведем пример:

```
<ol>
<li type=1> – арабские числа
<li type=1> – арабские числа
<li type=1> – арабские числа
<li type=a> – строчные буквы
<li type=a> – строчные буквы
<li type=a> – строчные буквы
<li type=A> – заглавные буквы
<li type=A> – заглавные буквы
<li type=A> – заглавные буквы
<li type=i> – малые римские цифры
<li type=i> – малые римские цифры
<li type=i> – малые римские цифры
<li type=I> – большие римские цифры
<li type=I> – большие римские цифры
<li type=I> – большие римские цифры
</ol>
```

Результат показан на рис. 1.17.

Списки определений **<DL> ... </DL>**

Список определений несколько отличается от других видов списков. Вместо меток `` в списках определений используются два уровня меток: метки `<DT>` (от английского definition term — определяемый термин) и `<DD>` (от английского definition definition — определение определения). Рассмотрим это на примере. Допустим, необходимо вывести фрагмент, представленный на рис. 1.18.

Для вывода на экран такого фрагмента нужно подготовить на языке HTML следующий документ:

```
<DL>
<DT><b>WEB-промоутер</b>
<DD>Web-промоутер это синтетическая специальность, <br>
включающая знание следующих дисциплин.... <DT>
```

HTML-документ

<DD>Текстовый файл с расширением *.htm
 или с расширением *.html, распознаваемый браузером

</DL>

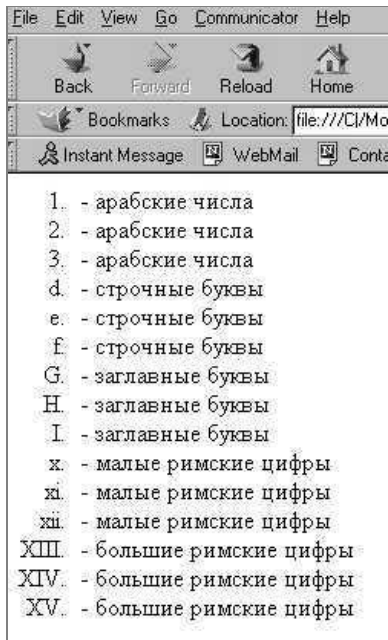


Рис. 1.17. Вариант смешанного списка

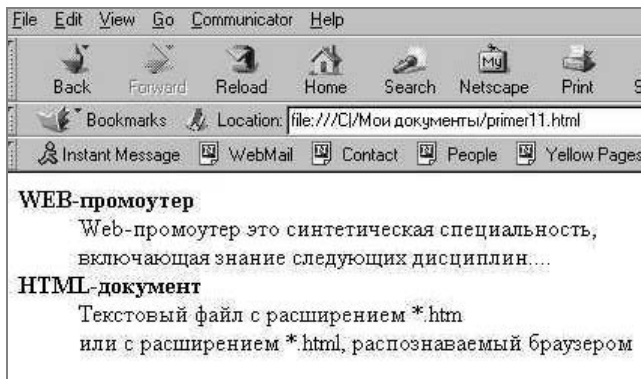


Рис. 1.18. Список определений

Обратите внимание на то, что так же, как и метки ``, метки `<DT>` и `<DD>` не имеют парных закрывающих меток, т. е. являются непарными. Если определяемые термины достаточно коротки, можно использовать модифицированную открывающую метку `<DL COMPACT>`.

Например, фрагмент:

```
<DL COMPACT>
<DT>А
<DD>Первая буква алфавита
<DT>Б
<DD>Вторая буква алфавита
<DT>В
<DD>Третья буква алфавита
</DL>
```

будет выведен на экран так, как показано на рис. 1.19.

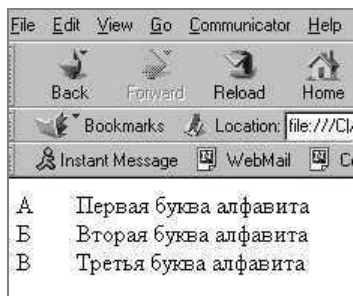


Рис. 1.19. Компактный список определений

Вложенные списки

Элемент любого списка может, в свою очередь, содержать в себе целый список любого другого вида. Число уровней вложенности в принципе не ограничено. Такая конструкция часто бывает удобна для оформления оглавлений отчетов, объемистых книг и диссертаций. Например:

```
<UL>
<li> первый элемент списка
<li> второй элемент списка
```

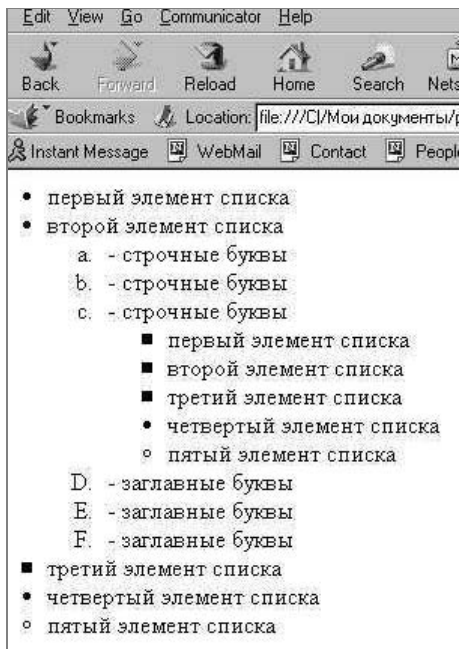


Рис. 1.20. Комбинированные вложенные списки

```
<ol>
```

```
<li type=a> – строчные буквы
```

```
<li type=a> – строчные буквы
```

```
<li type=a> – строчные буквы
```

```
<ul>
```

```
<li> первый элемент списка
```

```
<li> второй элемент списка
```

```
<li type=square> третий элемент списка
```

```
<li type=disc> четвертый элемент списка
```

```
<li type=circle> пятый элемент списка
```

```
</ul>
```

```
<li type=A> – заглавные буквы
```

```
li type=A> – заглавные буквы
```

```
<li type=A> – заглавные буквы
```

```
</ol>  
<li type=square> третий элемент списка  
<li type=disc> четвертый элемент списка  
<li type=circle> пятый элемент списка </ul>
```

На экране это отобразится так, как показано на рис. 1.20.

1.3. Организация таблиц

Таблицы нужны не только для представления определенного вида информации в табличном виде, но и являются также удобным инструментом структурирования всего документа в целом или его отдельной части. Однако следует подчеркнуть, что структурная табличная организация Web-сайта (в особенности его первой страницы) может существенно затруднить его продвижение (раскрутку).

Основные элементы таблицы

Проще всего создавать симметричные таблицы наподобие школьного листа в клеточку. Например такие, как таблица на рис. 1.21.



Название фильма	Страна	Год производства
Бриллиантовая рука	СССР	1971
Титаник	США	1998
Вор	Россия	1998

Рис. 1.21. Пример простой таблицы

Как создать такую таблицу? Сначала приведем соответствующий ей Web-документ, а затем сделаем необходимые пояснения.

```
<table border=3 align=left>  
<tr><th>Название фильма</th><th>Страна</th><th>Год  
производства</th></tr><tr>  
<td>Бриллиантовая рука</td><td>СССР</td><td>1971</td></tr>  
<tr><td>Титаник</td><td>США</td><td>1998</td></tr>  
<tr><td>Вор</td><td>Россия</td><td>1998</td></tr></table>
```

Таблица начинается с метки `<table>` и заканчивается меткой `</table>`. Метка `<table>` может включать несколько атрибутов:

Атрибут `BORDER` устанавливает ширину внешней рамки таблицы и ячеек в пикселах (например, `BORDER=3`). Если атрибут не установлен, таблица отображается без рамки.

Атрибут `ALIGN` устанавливает расположение таблицы по отношению к полям документа. Допустимые значения: `ALIGN=LEFT` (выравнивание влево), `ALIGN=CENTER` (выравнивание по центру), `ALIGN=RIGHT` (выравнивание вправо).

Атрибут `WIDTH` задает ширину таблицы. Ее можно указать в пикселах (например, `WIDTH=600`) или в процентах от ширины страницы (например, `WIDTH=75%`).

Атрибут `CELLSPACING` устанавливает ширину окантовки ячеек таблицы в пикселах (например, `CELLSPACING=5`).



Название фильма	Страна	Год производства
Бриллиантовая рука	СССР	1971
Титаник	США	1998
Вор	Россия	1998

Рис. 1.22. Таблица с ненулевыми значениями атрибутов `cellspacing` и `cellpadding`

Атрибут `CELLPADDING` устанавливает расстояние между рамкой ячейки и текстом в пикселах, т. е. дает, как говорят в полиграфии, "воздух" (например, `CELLPADDING=10`). Покажем, как влияют эти параметры на внешний вид таблицы. В предыдущем примере изменим лишь первую строчку: `<table border=3 align=left cellspacing=5 cellpadding=10>`. Получим таблицу, представленную на рис. 1.22.

Таблица может иметь заголовки (`<CAPTION> ... </CAPTION>`), хотя он и не является обязательным. Тег `<CAPTION>` может включать атрибут `ALIGN`. Допустимые значения: `<CAPTION ALIGN=TOP>`

(заголовок помещается над таблицей) и `<CAPTION ALIGN=BOTTOM>` (заголовок помещается под таблицей). По умолчанию заголовок таблицы помещается вверху (рис. 1.23).

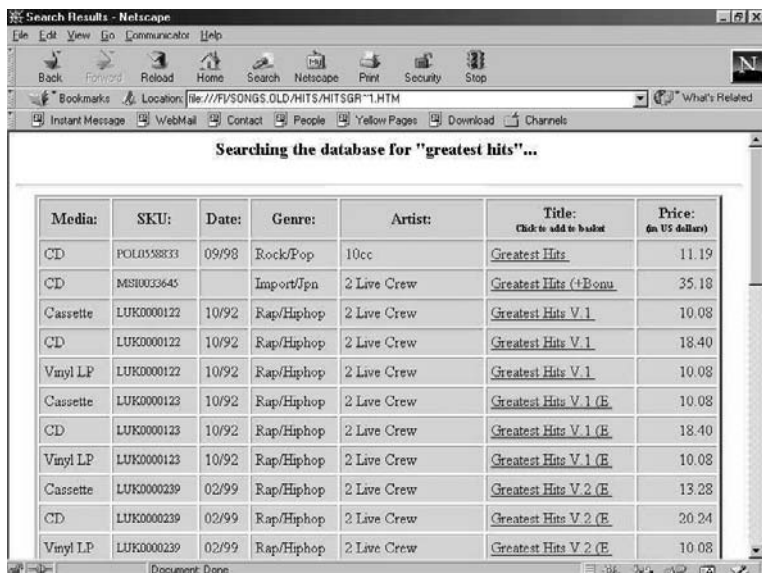


Рис. 1.23. Пример таблицы с заголовком

Каждая строка таблицы начинается с метки `<TR>` и заканчивается меткой `</TR>`. Метка `<TR>` может включать следующие атрибуты, действие которых распространяется на все ячейки данной строки:

□ Атрибут `ALIGN` устанавливает выравнивание текста в ячейках строки. Допустимые значения:

- `ALIGN=LEFT` (выравнивание влево)
- `ALIGN=CENTER` (выравнивание по центру)
- `ALIGN=RIGHT` (выравнивание вправо)

□ Атрибут `VALIGN` устанавливает вертикальное выравнивание текста в ячейках строки. Допустимые значения:

- Атрибут `ALIGN=TOP` (выравнивание по верхнему краю)
- Атрибут `VALIGN=MIDDLE` (выравнивание по центру)
- Атрибут `VALIGN=BOTTOM` (выравнивание по нижнему краю)

Каждая ячейка таблицы начинается с метки `<TD>` и заканчивается меткой `</TD>`. Метка `<TD>` может включать перечисленные ниже атрибуты, действие которых распространяется только на данную ячейку.

- Атрибут `nowrap`. Присутствие этого атрибута означает, что содержимое ячейки должно быть показано в одной строке.
- Атрибут `colspan` устанавливает "перекрытие" числа колонок ячейки по горизонтали. Например, `colspan=3` означает, что ячейка занимает три колонки.
- Атрибут `rowspan` устанавливает "перекрытие" числа рядов ячейки по вертикали. Например, `rowspan=2` означает, что ячейка занимает два ряда.
- Атрибут `align` устанавливает выравнивание текста в каждой отдельной ячейке. Допустимые значения:
 - `align=left` (выравнивание влево)
 - `align=center` (выравнивание по центру)
 - `align=right` (выравнивание вправо)
- Атрибут `valign` устанавливает вертикальное выравнивание текста в ячейке. Допустимые значения:
 - `valign=top` (выравнивание по верхнему краю)
 - `valign=middle` (выравнивание по центру)
 - `valign=bottom` (выравнивание по нижнему краю)
- Атрибут `width` устанавливает ширину ячейки в пикселах (например, `width=200`).
- Атрибут `height` устанавливает высоту ячейки в пикселах (например, `height=40`). Если ячейка таблицы пуста, вокруг нее не рисуется рамка. Если рамка нужна, то в ячейку можно ввести символьный объект ` ` (`non-breaking space` — неразрывный пробел). Ячейка по-прежнему будет пустой, а рамка вокруг нее останется.

Приведем пример составления таблицы сложной структуры:

```
<table border=3 align=center cellspacing=10 cellpadding=5
bgcolor="#e1dfda">
<tr><th>Названия</th><th>Параметры</th><th colspan=3>В
см/футах </th></tr>
```