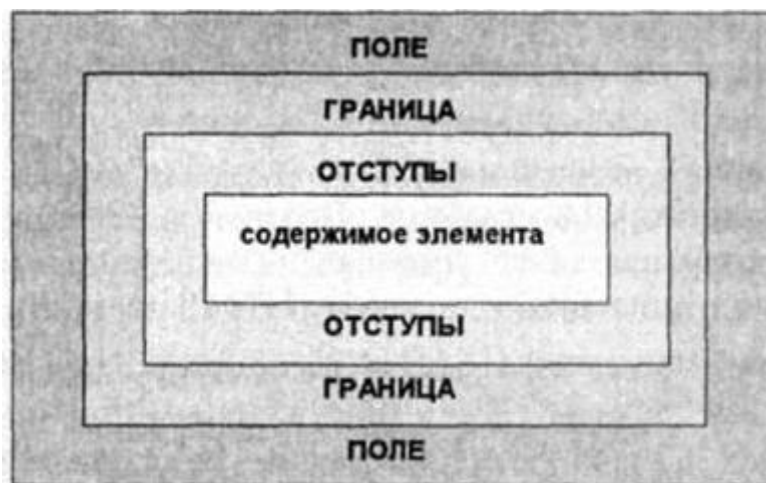


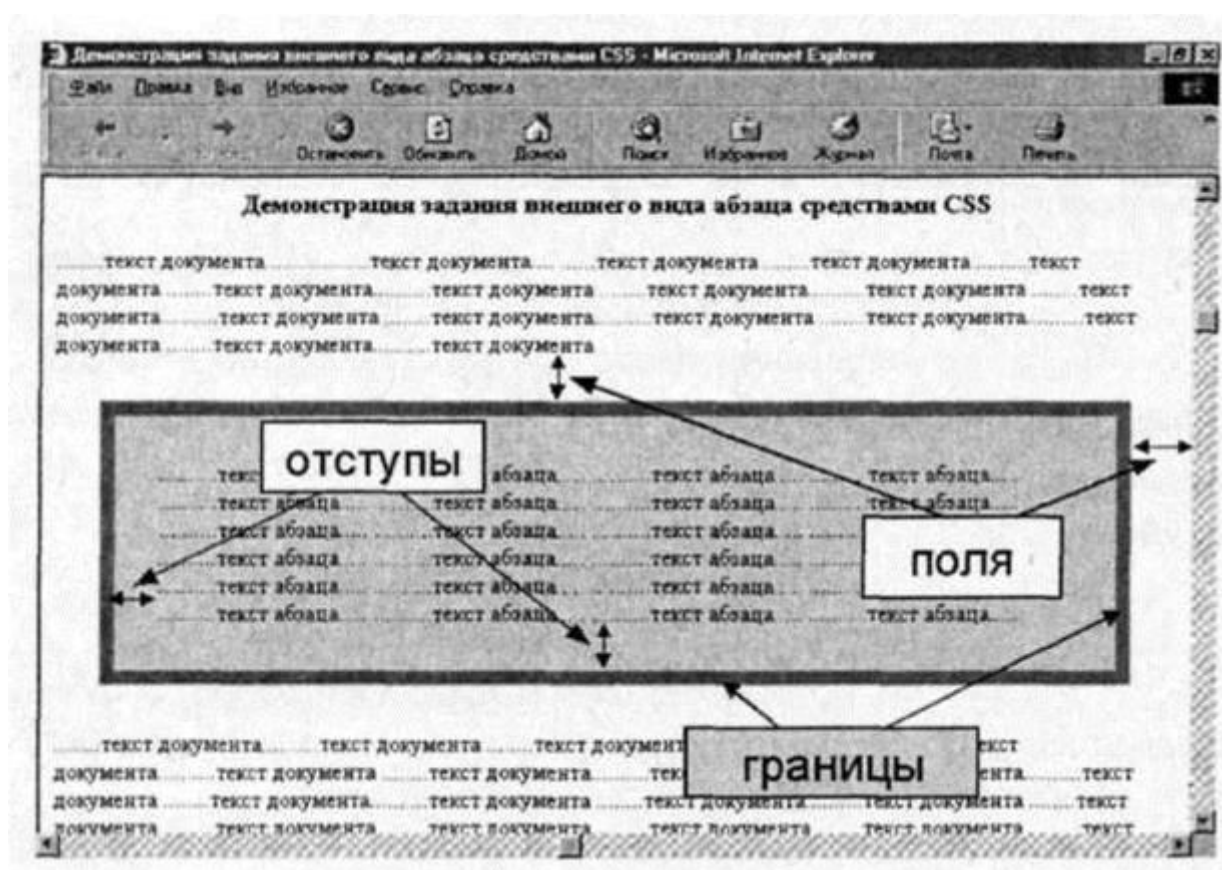
# Блочная модель визуального представления

Одной из основных технологий, составляющих суть языка CSS2, является блочная модель отображения документов. Согласно этой модели, html-элементы уровня блока, состоящие в иерархическом дереве документа, отображаются в виде прямоугольных блоков, к которым могут быть применены стилевые установки.



Сам блок представляет собой несколько вложенных друг в друга прямоугольных областей. В самой внутренней области и помещается информативное содержание html- элемента

---



Под html-элементы, являющиеся строковыми, стилевые блоки не выделяются. Например, согласно блочной модели документа, визуальное представление абзаца (элемента p) будет иметь вид.

При этом текст абзаца (как это можно видеть на рисунке), представляющий собой информативную область блока, имеет вокруг себя три прямоугольных области: отступ, границу и поле. Каждая из них может быть разбита на четыре сегмента: левый, правый, верхний и нижний. В дальнейшем левый сегмент области отступа будем называть левым отступом, правый сегмент области границы — правой границей и т.д. Возможности языка CSS2 позволяют для каждой области устанавливать свои визуальные параметры (цвет, фон, размер и т.д.). Стилиевое оформление фона различных областей определяется следующим образом:

- фоновое оформление информативной области задается CSS-свойствами html- элемента, начинающимися со слова background;
- область отступов также использует для своего фона свойства background html- элемента, породившего данный блок;
- область границы сама имеет свои свойства, устанавливающие ее фоновое оформление;
- область поля всегда является прозрачной и служит только для корректировки расположения блока на Web-странице. Поэтому в качестве цвета области поля используется цвет, установленный свойством background родительского по отношению к текущему html-элементу.

Как уже писалось в главе, посвященной html 4.01, универсальным элементом уровня блока является элемент div, универсальным строковым элементом — элемент SpAN. Смысл их использования заключается только в привнесении структуры в html-документ. Как элементы html, они не задают никаких визуальных параметров для своего содержимого. Информация между начальным и конечным тегами элементов div и SpAN считается блоком или строкой соответственно.

В рамках блочной модели представления документов использование этих элементов принимает особое значение. Например, благодаря элементу `div` можно выделить как блок группу строчковых элементов или фрагмент текста, что автоматически позволяет применять CSS-свойства, соответствующие блокообразующим элементам. То же самое можно было бы сделать с помощью элемента `p` (элемента абзаца), но иногда, с точки зрения смысловой организации структуры документа, было бы правильнее использовать универсальный элемент блока `div`. К тому же использование элемента абзаца приведет к выделению его содержимого пустыми строками сверху и снизу. То есть элемент `p`, как элемент с четко определенным смысловым значением (элемент `p` — элемент абзаца), привносит свойственные ему особенности визуального представления. Элемент `div` всего этого лишен, предоставляя разработчику самому определить его смысловое значение и особенности внешнего вида.

размеры прямоугольной области экрана, занимаемой стилевым блоком `html`-элемента, выделяются как сумма значений следующих свойств:

• горизонтальный размер:

левое поле (`margin-left`); + левая граница (`border-left`); + левый отступ (`padding-left`);

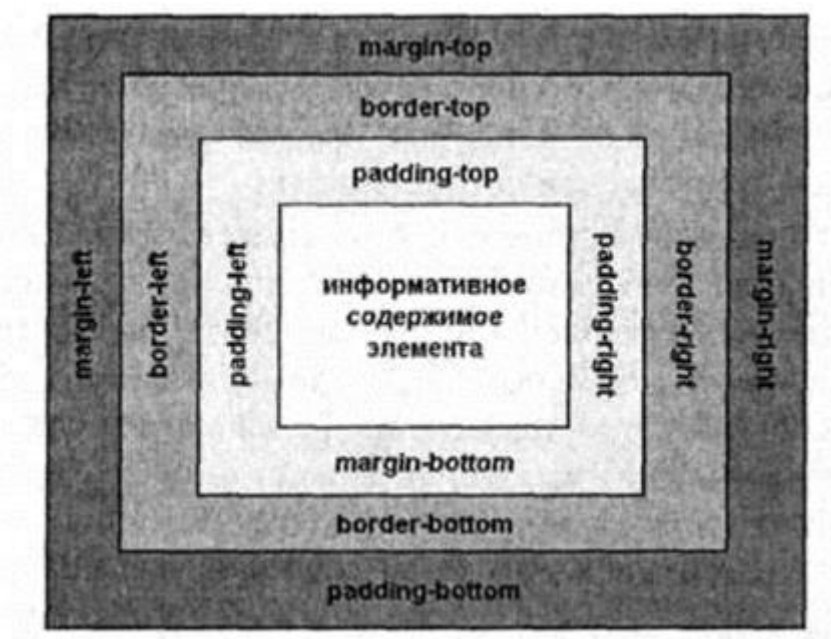
ширина информативной области (атрибут `width` блокообразующего элемента, который также может указываться в таблице стилей);

правый отступ (`padding-right`); + правая граница (`border-right`); + правое поле (`margin-right`).

• вертикальный размер вычисляется аналогично горизонтальному с той разницей, что вместо правых и левых (`left` и `right`) суммируются верхние и нижние (`top` и `bottom`) размеры соответствующих областей. Наглядно это представлено на рис. 5.10.

Надо отметить, что если два блока прилегают друг к другу, то размеры верхнего и нижнего полей указывают минимальное расстояние между границами соседних блоков. Поэтому если для двух прилегающих блоков задано ненулевое значение вертикальных полей, то они сливаются в одно с размером, равным наибольшей среди двух блоков высоте поля. Хороший пример такого слияния можно наблюдать при отображении элементов списка `LI`.

Вертикальные поля "плавающих" элементов (для списка которых задано свойство `float`) никогда не перекрываются.



Далее по тексту будет произведено подробное описание свойств различных областей стилевого блока.

# Свойства полей

Свойства полей задают геометрические размеры области поля стилевого блока. Возможности CSS2 позволяют задать размер каждого поля отдельно или для всех сразу. Следующие свойства задают размеры одного отдельного поля:

- `margin-left`: размер левого поля;
- `margin-right`: размер правого поля;
- `margin-top`: размер верхнего поля;
- `margin-bottom`: размер нижнего поля.

размеры указываются в стандартных значениях длины, определенных для языка CSS2, или в процентах ширины ближайшего блокового элемента-родителя. Пример:

```
<html>

<head>

<title>Пример внедренной таблицы стиля </title>

<style type = "text/css"> p {margin-left: 10 px; margin-right: 10 px; margin-top: 12 px }

</style>

</head>

<body>

    .текст документа

<p>

    .текст абзаца

    .текст абзаца

</p>

    .текст документа

</body> </html>
```

Как видно из примера, необязательно указывать размеры всех четырех полей стилевого блока. При этом для всех неопределенных полей будет использоваться значение, используемое по умолчанию, т.е. нулевой размер поля.

Для задания размеров сразу всех полей может быть использована сокращенная (стенографическая) форма их указания в виде свойства `margin`. При этом через пробел указываются размеры всех четырех полей. Например,

```
p {margin: 12px 10px 0px 16px}
```

Обратите внимание на порядок, в котором задаются и интерпретируются браузером размеры полей: сначала указывается размер верхнего поля, затем размер правого поля, затем размер нижнего поля и, наконец, размер левого поля.

### Свойства отступов

Геометрические размеры отступов, так же как и в случае с полями, можно задать либо каждым отдельным правилом, либо все сразу. Ниже перечислены свойства, задающие размеры каждого поля в индивидуальном порядке:

padding-left: размер левого отступа; padding-right: размер правого отступа; padding-top: размер верхнего отступа; padding-bottom: размер нижнего отступа.

размеры также указываются в стандартных для CSS2 значениях длины или в процентном

задании от ширины ближайшего родительского элемента. Сокращенная форма задания размеров всех четырех отступов осуществляется с помощью правила padding. Например:

```
pRE {background: lightgrey; font-color: blue;  
  
padding: 10px 15px 10px 15px}
```

размеры отступов задаются и интерпретируются браузером в следующем порядке: сначала указывается размер верхнего отступа, затем размер правого отступа, затем размер нижнего отступа и, наконец, размер левого отступа.

### Свойства границ

Границы представляют собой обыкновенные линии, ограничивающие видимую часть стилевого блока (поля являются прозрачными и поэтому невидимы). Определенные в CSS2 свойства позволяют задавать толщину, цвет и тип этих линий.

#### Ширина границ

Ширина границ задается либо для каждой из них индивидуально, либо для всех четырех сразу. Ширина отдельных границ задается следующими свойствами:

border-top-width: ширина верхней границы; border-bottom-width: ширина нижней границы; border-right-width: ширина правой границы; border-left-width: ширина левой границы.

Ширина всех четырех границ может быть указана с помощью одного свойства border- width. Значения этих свойств либо указываются только в стандартных для CSS2 единицах длины, либо в качестве значений используются специально зарезервированные ключевые слова языка CSS2:

thin ? — тонкая линия границы; medium ? — средняя линия границы; thick ? — толстая линия границы.

Следует отметить, что не существует общепринятого для всех браузеров числового эквивалента ключевым словам. Поэтому в разных браузерах ширина границы, заданная одним и тем же ключевым словом, может быть разной.

Единственное, что можно сказать со всей уверенностью, так это то, что ширина границы thin меньше ширины границы medium, которая, в свою очередь, меньше ширины границы thick. Или, короче говоря,

```
thin<medium<thick>
```

Пример задания границ:

```
div (padding : 10 10 10 10; border-top-width: 5px; border-bottom-width: 5; border-right-width: thin; border-left-width: thin )
```

Или то же самое, но с использованием стенографического свойства border-width:

```
div {padding : 10px 10px 10px 10px; border-width: 5px thin 5px thin}
```

```
<html>

<head>

<title>
```

Порядок задания ширин границ для свойства border-width следующий: ширина верхней границы, ширина правой границы, ширина нижней границы, ширина левой границы. Все значения, как и в предыдущих случаях, разделяются пробелами. Пример использования (рис. 5.11).

Демонстрация рамок, толщина которых задана ключевыми словами thin medium и thick

```
</title>

<style type="text/css"> div { margin:20; padding:20;

text-align:center; border-style:solid; border-color:black; border-width:thin}

div.2 (border-width:medium} div.3 {border-width:thick}

</style>

</head>

<body bgcolor=white>

<h3 align=center>
```

Демонстрация рамок, толщина которых задана ключевыми словами

```
</h3>
```

```
<div>
```

Границы этого блока имеют толщину thin

```
</div>
```

```
<div class="2">
```

Границы этого блока имеют толщину medium

```
</div>
```

```
<div class="3">
```

Границы этого блока имеют толщину thick

```
</div>
```

```
</body>
```

```
</html>
```

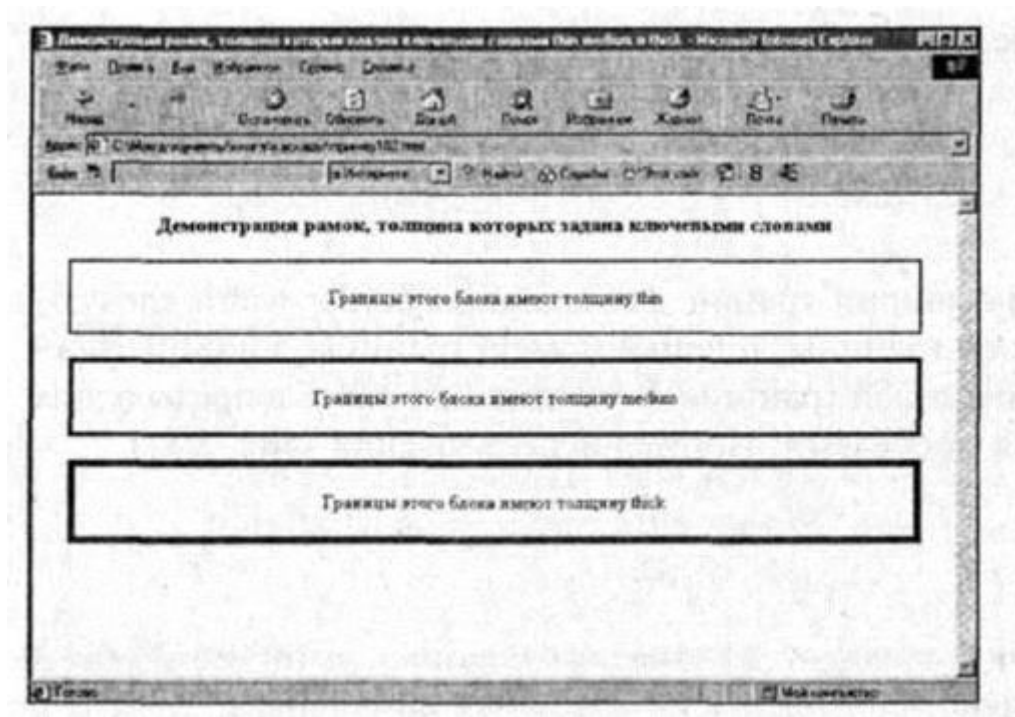


рис. 5.11. Вид рамок с различными значениями border-width в браузере Internet Explorer

Вид, представленный на рис. 5.11, вышеприведенный документ будет иметь только в браузере Internet Explorer. Netscape Communicator 4.74 не поддерживает селектор классов, и поэтому в его окне все границы будут иметь толщину thin. Для того чтобы продемонстрировать другие две границы (medium и thin) в Netscape Communicator 4.74, надо немного изменить документ:

```
<html>
```

```

<head>

<title>

Демонстрация рамок, толщина которых задана ключевыми словами thin medium и thick

</title>

<style type="text/css"> div {margin:20; padding:20;

text-align:center; border-style:solid; border-color:black; border-width:thin}

</style>

</head>

<body bgcolor=white>

<h3 align=center>

Демонстрация рамок, толщина которых задана ключевыми словами

</h3>

<div>

Границы этого блока имеют толщину thin

</div>

<div style="border-width:medium"> Границы этого блока имеют толщину medium

</div>

<div style="border-width:thick"> Границы этого блока имеют толщину thick

</div>

</body>

</html>

```

Такой документ отображается одинаково и браузером Internet Explorer и браузером Netscape Communicator. Это является простейшим примером создания браузеронезависимых документов. В данном случае все обошлось небольшой поправкой, но зачастую приходится отказываться от утрjn]hs[возможностей для того, чтобы документ имел одинаковый вид и в Internet Explorer и в Netscape Communicator.

---



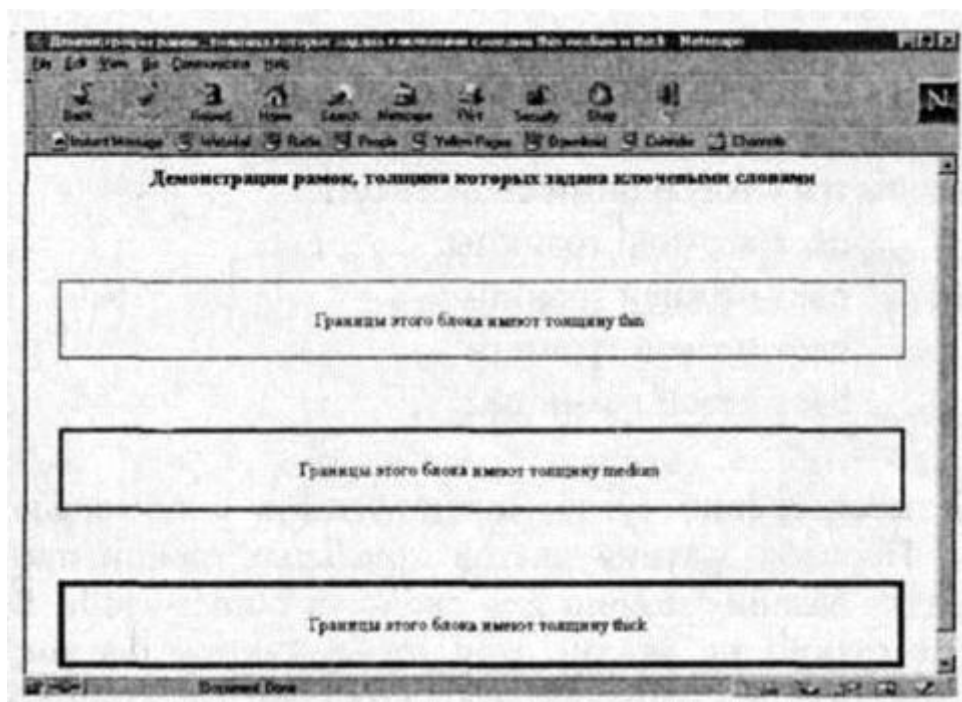


рис. 5.12. Вид рамок с различными значениями

border-width в браузере Netscape Communicator

Кстати, из рис. 5.12 можно видеть, что при отображении прилегающих блоков браузер Netscape Communicator 4.74, в отличие от Internet Explorer, суммирует поля между ними. Примечательно, что делает он это с блоками, имеющими границы. Если соседние блоки не имеют границ, то поля отображаются так, как надо: между блоками используется одно поле, размер которого больше. Например, если в рассматриваемом примере убрать рамки, то документ в окне Netscape Communicator 4.74 примет вид, представленный на рис. 5.13. К Netscape Communicator 6 это не относится, он отображает все так, как нужно.

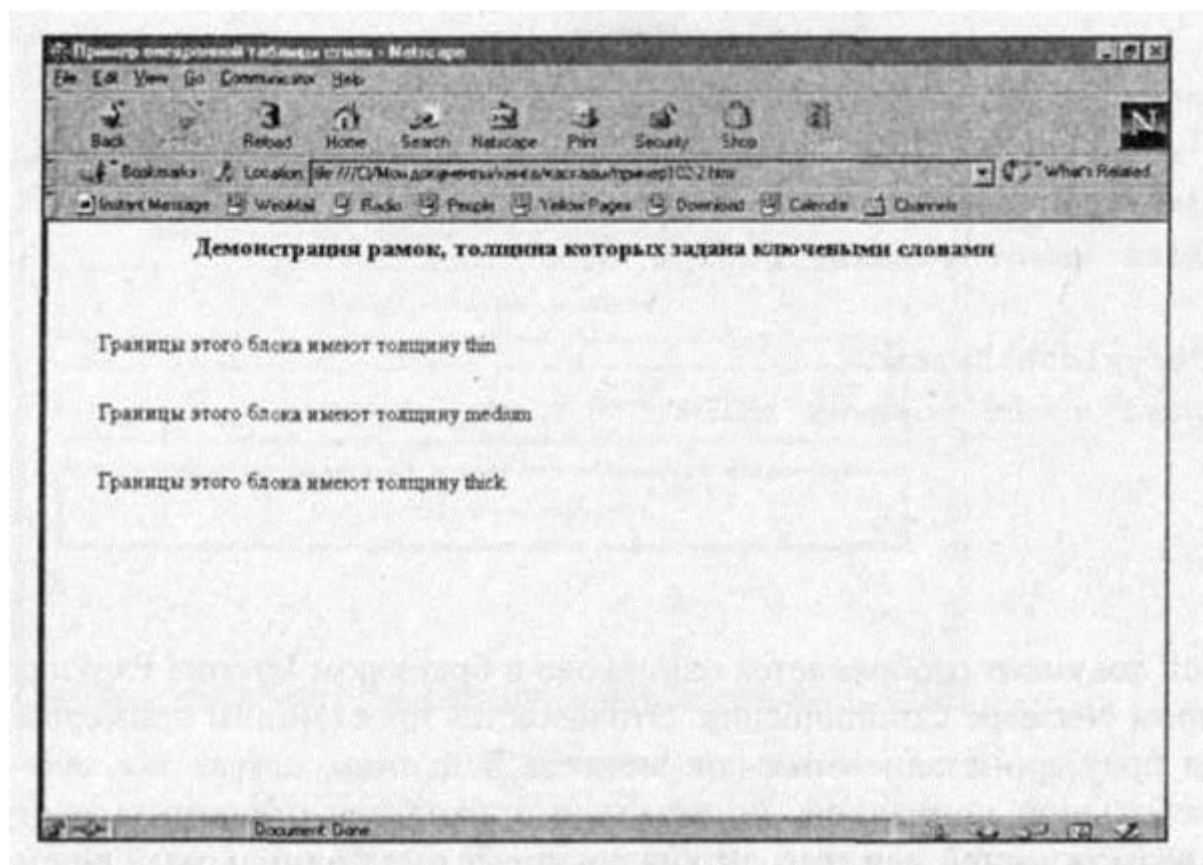


рис. 5.13. Отображение блоков с убранными границами в окне браузера Netscape Communicator 4.74

### Цвет границ

Цвет границ определяется следующими свойствами: `border-top-color`: цвет верхней границы; `border-bottom-color`: цвет нижней границы; `border-right-color`: цвет правой границы; `border-left-color`:

цвет левой границы.

Цвет для всех четырех границ сразу можно указать с помощью свойства `border-color`. Порядок задания цветов отдельных границ при этом аналогичен порядку задания ширин для свойства `border-width`. В том случае, если цвет границ не указан, они отображаются цветом, установленным для свойства `color` блокообразующего элемента. Пример задания стиля абзаца с указанием цвета границ:

```
p {padding: 10px 10px 10px 10px; border-width: 5px thin 5px thin; border-color: blue}
```

### Тип линии границ

Для границ должен быть также задан тип линии, которой она отображается. Ниже приведены допустимые в CSS2 названия типов и описан соответствующий им видимый результат:

`none` граница определена как отсутствующая. При таком задании значение

`border-width` устанавливается равным нулю;

`solid` граница будет отображаться в виде одной сплошной линии. Если

атрибут `border-width` не задан, то по умолчанию используется значение `border-width: medium`;

double граница будет очерчена двумя сплошными линиями. Причем свойство border-width

будет показывать суммарную толщину двух этих линий и расстояния между ними;

dotted граница будет отображена в виде пунктирной линии;

groove граница будет показана как вдавленная линия;

ridge . граница будет отображены в виде выпуклой линии;

inset при задании такого типа границ, весь блок будет отображен как

вдавленный;

outset указание этого значения при задании типа границ приведет к тому,

что весь блок будет отображен как выпуклый.

Тип границы блока может быть задан с помощью определения следующих свойств:

**border-top-style:** задает тип верхней границы; **border-bottom-style:** задает тип нижней границы;  
**border-left-style:** задает тип левой границы; **border-right-style:** задает тип правой границы.

Или с помощью одного стенографического свойства border-style могут быть указаны типы всех четырех границ сразу. Пример задания:

```
<html>
```

```
<head>
```

Порядок задания типов линий границ для свойства border-style идентичен порядку, определенному для всех предыдущих аналогичных свойств: верхняя граница, правая, нижняя и левая. Пример использования (рис. 5.14 и 5.15):

```
<title>Примеры рамок разного стиля</title>
```

```
<style type="text/css"> div{margin:5; padding:10;
```

```
text-align:center; border-style:solid; border-color:black; border-width:medium
```

```
)
```

```
</style>
```

```
</head>
```

```
<body bgcolor=white>
```

```
<h3 align=center>Демонстрация рамок разного стиля</h3>
```

```

<div>

Границы этого блока имеют стиль solid

</div>

<div style="border-style:double"> Границы этого блока имеют стиль double

</div>

<div style="border-width:15; border-style:groove">

Границы этого блока имеют стиль groove </div> <div style="border-width:15;
border-style:ridge"> Границы этого блока имеют стиль ridge </div> <div style="border-width:15;
border-style:inset">

Границы этого блока имеют стиль inset </div> <div

style="border-width:15; border-style:outset">

Границы этого блока имеют стиль outset </div> </body>

</html>

```

Как видно из рис. 5.15, браузер Internet Explorer 5.0 поддерживает границы стиля solid и double и игнорирует все остальные. Зато границы стиля double выглядят в нем гораздо приятнее, чем в Netscape Communicator (рис. 5.14).

Возможности языка CSS2 позволяют, кроме всего прочего, использовать другие формы описания внешнего вида границ. В общем случае для каждой границы указывается три визуальных параметра: цвет, толщина и тип линии. Поэтому в язык CSS2 включены свойства, определяющие их вместе для каждой из границ:

border-top: задает ширину верхней границы, тип линии и ее цвет;



рис. 5.14. Демонстрация границ разного стиля в браузере Netscape Communicator



рис. 5.15. Демонстрация границ разного стиля в браузере Internet Explorer 5.0

`border-bottom:`, `border-left:`, `border-right:` задают вышеперечисленные характеристики для нижней, левой и правой границ соответственно. Пример использования:

```
h3 {border-bottom: 10px solid blue}
```

Благодаря этому правилу все заголовки третьего уровня будут подчеркнуты, синей одинарной линией. Следует помнить о том, что если для свойства, например, `border-top`, указаны не все три визуальные характеристики границы, то для недостающего параметра будет использоваться значение, установленное по умолчанию.

Например:

```
pRE { border-color: green; border-top: double; border-bottom: double}
```

В этом случае для элемента `pRE` задан зеленый цвет всех границ. Однако верхняя и нижняя границы будут показаны черным цветом (цветом, используемым по умолчанию для вывода текста). Это произойдет потому, что свойства `border-top` и `border-bottom`, помимо типа линии границы, определяют ее цвет и толщину. Так как цвет разработчиком не указан, то эти свойства используют цвет, установленный для элемента `pRE` по умолчанию, то есть черный. А поскольку свойства `border-top` и `border-bottom` по тексту заданы после свойства `border-color`, то они переопределяют

его значение. Чтобы этого не происходило, достаточно следующим образом поменять их местами: pRE { border-top: double;

border-bottom: double; border-color: green}

```
<html>

<head>

<title>
```

В этом случае верхняя и нижняя границы будут отображены зеленой двойной линией.

Правая и левая границы не будут показаны вовсе.

Самым обобщенным свойством границ является CSS-свойство border. Оно задает ширину, тип линии и цвет сразу для всех границ. Причем для каждого параметра указывается одно значение, которое применяется ко всем четырем границам.

Интереснее всего обстоит дело с заданием полей, границ и отступов для элемента body. Браузер Netscape Communicator при этом воспринимает его как обычный блокообразующий элемент и обращается с ним соответствующим образом. рассмотрим это на примере. В окне Netscape Communicator приведенный ниже документ будет иметь вид, как на рис. 5.16.

Пример задания свойств полей, границ и отступов для элемента body

```
</title>

<style type="text/css"> body{margin:20; padding:20;

text-align:justify ; border-color:darkgray ; border-width:50

}

</style>

</head>

<body bgcolor=white>

<h3 align=center>Забавные музыканты</h3>
```

В музыке каждый понимает по мере веса слона, в свое время наступившего ребенку на ухо. Президент США Улисс Симпсон Грант, тот, который с полтинника, вообще говорил, что знает только две мелодии, одна из которых - "Янки дудл". А на вопрос, какая же вторая, гордо отвечал: "Не "Янки дудл". Впрочем, лучше, когда руководитель так разбирается в музыке, чем, скажем, как Жданов, который учил композиторов, какая должна быть музыка - чтоб ему, Жданову, было приятно ее напевать. Лучше бы Жданов пил, что ли - на Гранта еще в бытность его генералом за алкоголизм самому Линкольну стучали. Тот, правда, отреагировал весьма своеобразно - велел разузнать, какой сорт виски Грант пьет, чтоб и прочим генералам послать за свой счет по бочке, ибо



воевал-то Грант неплохо. Так что лучше, чтоб руководитель своих музыкальных вкусов не демонстрировал. разве что как Елизавета Английская, даровавшая некому доктору Джону Булю(интересное,кстати,совпадение) герб с надписью "Sol, mi, re, fa" - за сочинение английского гимна "Боже', храни королеву". А то вот Николай I не любил Глинку и разрешал заменять провинившимся офицерам гауптвахту посещением оперы "руслан и Людмила".

<p align=right>Ворос Бурда</p>

</body>

</html>

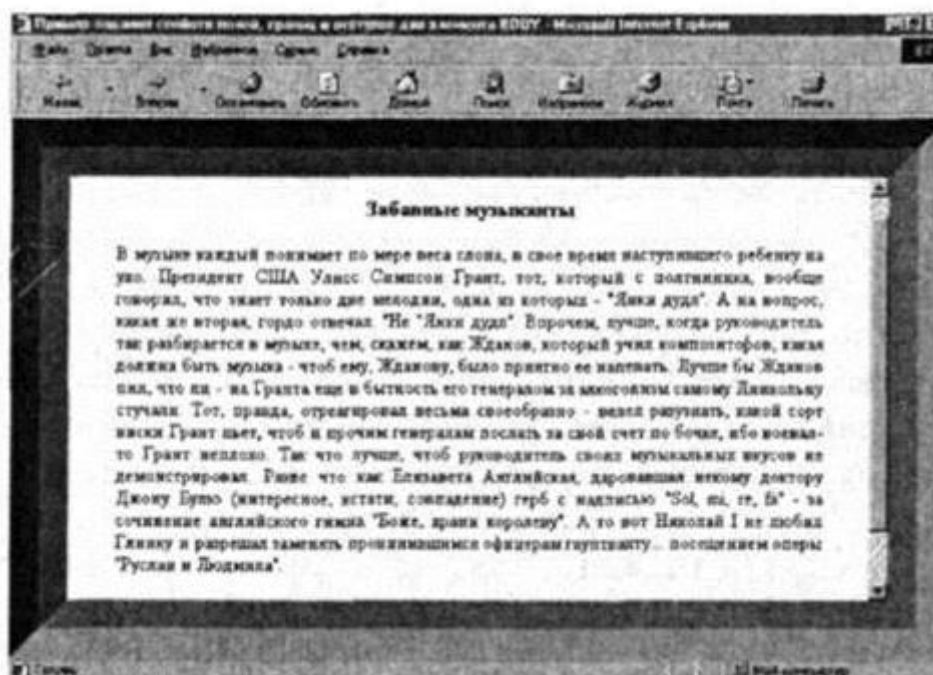


рис. 5.16. Пример документа с заданными размерами полей, границ и отступов элемента body, открытого в браузере Netscape Communicator

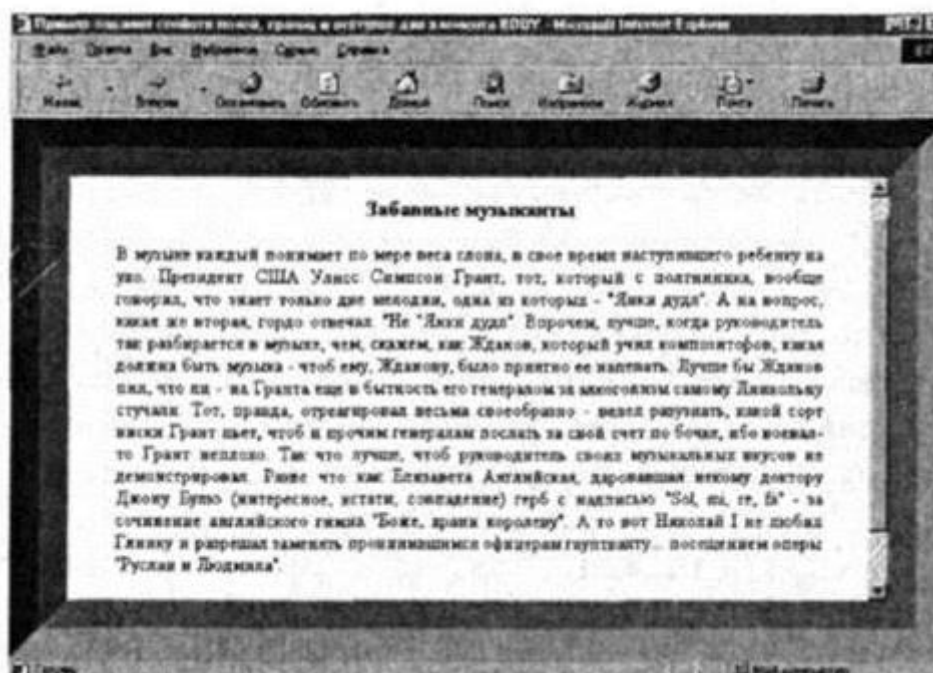


рис. 5.17. Пример документа с заданными размерами полей, границ и отступов элемента body, открытого в браузере Internet Explorer

Самое интересное начинается при открытии этого документа в браузере Internet Explorer. При этом для элемента body поля и границы меняются местами. Таким образом, у стилевого блока элемента body сначала идут границы, затем поля и потом отступы. Причем по умолчанию для элемента body браузер Internet Explorer использует оригинальный стиль границ. Для границ элемента body браузер Internet Explorer может использовать только стиль, установленный по умолчанию, или стиль single.

## Задание цвета текста и фона

Как уже упоминалось, каскадные таблицы стилей были задуманы для наиболее эффективного задания внешнего вида содержимого html-элементов, который они принимают в окне браузера. А что же, как не цветовое оформление является основной визуальной характеристикой. Для каждого отображаемого html-элемента может быть указано два цвета: цвет фона и цвет переднего плана. На переднем плане отображается информативное содержимое элемента, проще говоря — текст. В CSS2 цвет текста задается единственным, предназначенным для этих целей свойством color. Цвет указывается одним из стандартных, определенных в CSS2 способов: ключевым словом или RGB-функцией, например:

```
p {color: olive}
```

```
B {color: RGB (255,0,0)}
```

Для описания фонового оформления элемента в CSS2 определено несколько возможностей.

При этом фон можно задать с помощью следующих свойств: background-color: указывает цвет фона.

background-image: указывает фоновое изображение. Причем для этого случая может быть задано несколько дополнительных параметров фона:

- о background-repeat: указывает наличие или отсутствие дублирования фонового изображения как по вертикали, так и по горизонтали;

- о background-attachment: это свойство указывает на то, будет или нет фоновое изображение прокручиваться при прокрутке документа;

- о background-position: задает положение изображения в рамках стилевого блока, к которому оно принадлежит.



## Свойство background-color

Использование этого свойства аналогично использованию свойства color. разница заключается в том, что для строковых элементов свойство color, определяет цвет текста, а background-color — цвет фона, на котором этот текст отображается. Свойство background-color заданное для блокообразующего элемента, указывает цвет, которым будет залито все информационное пространство стилевого блока вместе с его отступами. Пример использования (рис. 5.18):

```
<html>

<head>

<title>

Пример задания свойств полей, границ и отступов для элемента body

</title>

<style type="text/css"> body{margin:20; padding:20;

text-align:justify; border-color:darkgray; border-width:50

)

</style>

</head>

<body bgcolor=white>

<h3 align=center>Забавные музыканты</h3>

В музыке каждый понимает по мере веса слона, в свое время наступившего ребенку на ухо. Президент США Улисс Симпсон Грант, тот, который

с полтинника, вообще говорил, что знает только две мелодии, одна из

которых - "Янки дудл". А на вопрос, какая же вторая, гордо отвечал: "Не "Янки дудл". Впрочем, лучше, когда руководитель так разбирается

в музыке, чем, скажем, как Жданов, который учил композиторов, какая должна быть музыка - чтоб ему, Жданову, было приятно ее напевать. Лучше бы Жданов пил, что ли - на Гранта еще в бытность его генералом за алкоголизм самому Линкольну стучали. Тот, правда, отреагировал весьма своеобразно - велел разузнать, какой сорт виски Грант пьет, чтоб и прочим генералам послать за свой счет по бочке, ибо воевал-то Грант неплохо. Так что лучше, чтоб руководитель своих музыкальных вкусов не демонстрировал. разве что как Елизавета Английская, даровавшая некому доктору Джону Булю (интересное, кстати, совпадение) герб с надписью "Sol, mi, re, fa" - за сочинение английского гимна "Боже, храни королеву". А то вот Николай I не любил Глинку и разрешал заменять провинившимся офицерам гауптвахту посещением оперы "руслан и Людмила". <p align=right>Борис Бурда</p> </body> </html>

! [img] (/clip_image021.jpg)
```

рис. 5.18. Пример документа, в оформлении которого использовалось свойство background-color

### ### Свойство background-image

В качестве значения этого свойства указывается URL-адрес изображения, которое будет использоваться в качестве фонового, например,

```
body (background-image: URL(my imagebg image.jpg))
```

```
p (color: yellow;
```

```
background-image: URL (my imageaaa32.jpg))
```

Для каждого блокообразующего html-элемента изображение будет занимать всю информативную область блока вместе с его отступами. Если отводимое под изображение пространство стилового блока по своим размерам меньше самого изображения, то, значит, последнее будет отображено частично. Геометрические размеры фоновой картинке никак не влияют на размеры стилового блока.

Примечательно, что фоновое изображение может быть задано и для строкового html- элемента. В этом случае отображается только та часть картинки, которая необходима для вывода текста. Ее левый верхний угол соответствует первой букве текста строкового элемента, например (рис. 5.19):

```
<html> <head> <title>
```

Пример использования свойства background-image для строковых элементов

```
</title>
```

```
<style type="text/css"> SpAN (background-image:URL(466009. jpg) } </style>
```

```
<body bgcolor=white>
```

```
<h3 align=center>Забавные политики</h3> <SpAN><B>
```

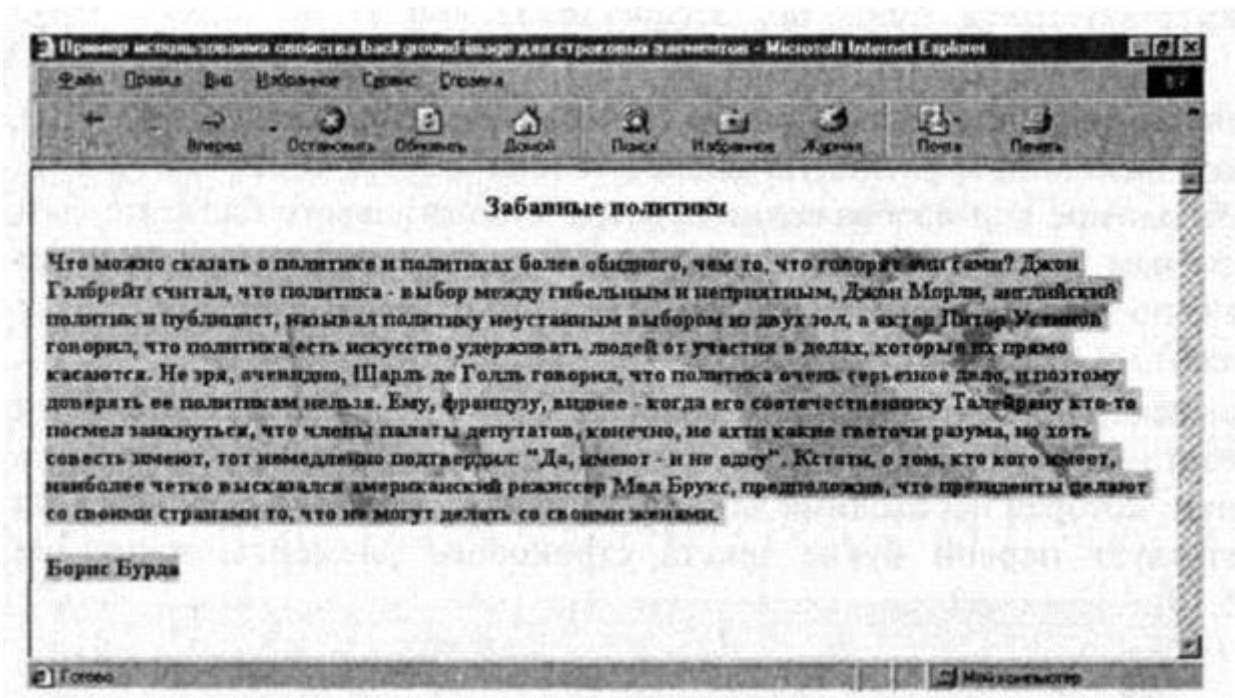
Что можно сказать о политике и политиках более обидного, чем-то, что говорят они сами? Джон Гэлбрейт считал, что политика - выбор между гибельным и неприятным, Джон Морли, английский политик и публицист, называл политику неустанным выбором из двух зол, а актер Питер Устинов говорил, что политика есть искусство удерживать людей от участия в делах, которые их прямо касаются. Не зря, очевидно, Шарль де Голль говорил, что политика очень серьезное дело, и поэтому доверять ее политикам нельзя. Ему, французу, виднее - когда его соотечественнику Талейрану кто-то посмел заикнуться, что члены палаты депутатов, конечно, не ахти какие светочи разума, но хоть совесть имеют, тот немедленно подтвердил: "Да, имеют - и не одну". Кстати, о том, кто кого имеет, наиболее четко высказался американский режиссер Мел Брукс, предположив, что президенты делают со своими странами то, что не могут делать со своими женами.

```
</SpAN>
```

```
<BR><BR>
```

```
<SpAN>Борис Бурда</SpAN>
```

```
</body>
```



## Свойство *background-repeat*

Это свойство используется, если для элемента задано фоновое изображение. Оно определяет, будет ли это фоновое изображение дублироваться, и если будет, то каким образом. По умолчанию фоновая картинка будет в виде мозаики заполнять все пространство стилевого блока, отведенное под его информативную область и отступы. Свойство `background-repeat` позволяет указать иной способ дублирования. Для этого оно может принимать следующие значения:

`repeat` изображение дублируется в вертикальном и горизонтальном направлении (используется

по умолчанию);

`repeat-x` изображение дублируется только в горизонтальном направлении;

`repeat-y` изображение дублируется только в вертикальном направлении;

`no-repeat` . изображение не дублируется вовсе. При этом оно будет отображено

только один- единственный раз. Пример использования (рис. 5.20):

```
<html>

<head>

<title>

Примеры различного задания свойства background-repeat

</title>

<style type="text/css">

p {background-image:URL(165.gif); margin:10;

padding:40; border:solid gray thin; text-align:center}
```

```

</style>

</head>

<body bgcolor=white>

<h3 align=center>Примеры различного задания свойства background repeat </h3>

<p style="background-repeat:norepeat">

Для этого блока <B>background-repeat:norepeat</B>

</p>

<p>

Свойство background-repeat для этого блока используется в значении, установленном по умолчанию.
То есть <B>background-repeat:repeat</B> </p>


<p style="background-repeat:repeat-x">

Для этого блока <B>background-repeat:repeat-x</B>

</p>

<p style="background-repeat:repeat-y">

Для этого блока <B>background-repeat:repeat-y</B>

</p>

</body>

</html>

```

Выложить фоновым изображением крест, дважды используя свойство background-repeat (один раз в значении repeat-x, другой раз в значении repeat-y) не получится. Объясняется это тем, что оба этих свойства являются одновременными, и поэтому второе просто-напросто переопределяет первое. Положение исходного изображения задается свойством background-position.

## *Свойство background-attachment*

Свойство background-attachment устанавливает, будет ли фоновое изображение фиксироваться в окне браузера или будет прокручиваться вместе с документом в процессе его прокрутки. Это свойство имеет смысл только для фоновых изображений элементов, для которых предусмотрены полосы прокрутки (элементы body и FRAME). Если изображение указано как зафиксированное, то содержимое элемента как бы проплывает над ним при прокрутке. В противном случае фоновое изображение (обычно продублированное) будет прокручиваться вместе с документом. Для задания параметров фиксации используются следующие значения свойства background-attachment:

fixed — фоновое изображение задается как зафиксированное;

scroll — фоновое изображение будет прокручиваться вместе с документом. По умолчанию используется значение scroll.

Пример задания:

```
body {background-image: url (gb706.gif); background-repeat : no repeat;  
  
background-position: center; background-attachment: fixed}
```

В этом примере изображение gb706.gif, используемое в качестве фона, будет отображено в единственном экземпляре (никакого дублирования), расположено по центру и зафиксировано.

## *Свойство background-position*

Это свойство своим значением определяет начальное месторасположение фонового изображения в рамках стилевого блока. Другими словами, оно показывает, где будет отображен первый экземпляр фонового изображения. Именно от него и будет производиться дублирование картинки в разные стороны.

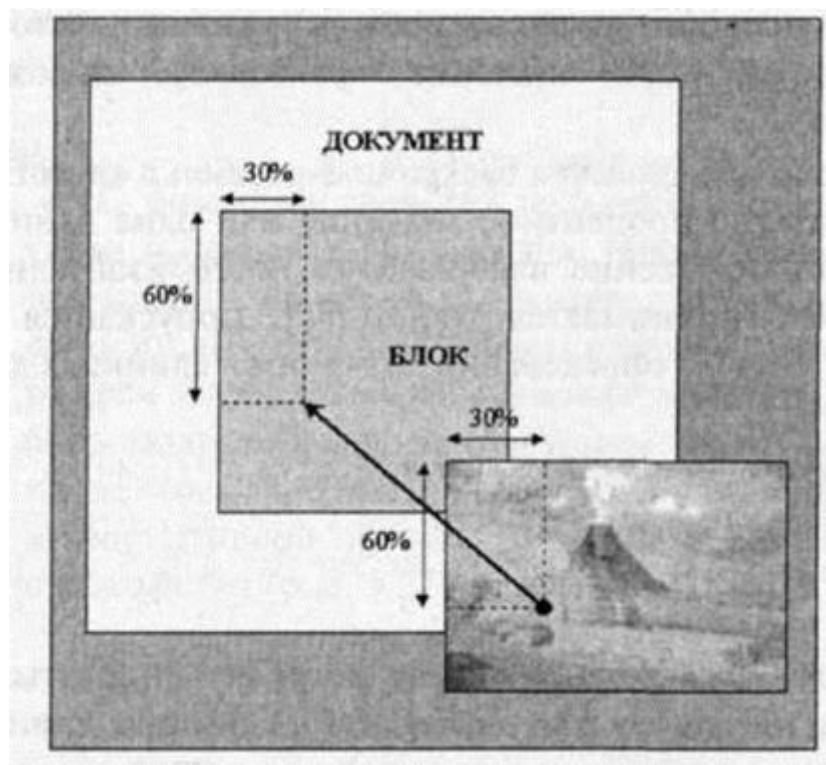
Через свойство background-position положение фоновой картинки может быть задано тремя способами.

### 1. В процентах.

В этом случае возможны следующие варианты позиционирования:

если в качестве значения свойства background-position указана пара 0% 0% (background-position: 0% 0%), то левый верхний угол изображения помещается в левый верхний угол области отступов;

если указана пара 30% 60%, то сначала браузером определяется точка изображения, полученная в результате смещения от левого верхнего угла изображения на 30% ширины вправо и на 60% высоты вниз. Затем изображение размещается в стилевом блоке таким образом, чтобы эта точка была совмещена с точкой блока, полученной в результате смещения вдоль области отступов на 30% вправо и на 60% вниз от верхнего левого угла области отступов;



если указано значение 100% 100%, то в этом случае правый нижний угол изображения совмещается с правым нижним углом области отступов стилевого блока.

## 2. В единицах длины.

Здесь все просто. Пара значений, например 20px 15px, показывает, что левый верхний угол изображения будет смещен на 20 пикселей вправо относительно левого верхнего угла области отступов и на 15 пикселей вниз.

## 3. С помощью ключевых слов top, center, bottom, right и left.

В этом случае положение фоновой картинки может быть задано одной из следующих комбинаций.

top left, top left комбинация аналогична процентному значению 0% 0%.

Обратите внимание на то, что пунктуация должна быть именно такой, какой она указана в книге.

bottom right, right bottom комбинация аналогична процентному значению 100% 100%.

center, center center комбинация аналогична процентному значению 0% 0%.

top, top center, center top комбинация аналогична процентному значению 50% 0%.

left, left center, center left комбинация аналогична процентному значению 0% 50%.

right top, top right комбинация аналогична процентному значению 100% 0%.

bottom left, left bottom комбинация аналогична процентному значению 0% 100%.

right, right center, center right комбинация аналогична процентному значению 100% 50%.

bottom, bottom center, center bottom комбинация аналогична процентному значению 50% 100%.

Здесь перечислены все возможные варианты использования ключевых слов. Для всех способов задания сначала указывается месторасположение по горизонтали, второе значение характеризует положение по вертикали.

В том случае, если для свойства `background-position` в качестве значения указано только одно процентное значение или одна длина, то это значение определяет положение изображения по горизонтали. Вертикальная составляющая принимается равной 50%. Допускается совместное использование в одном определении значений в единицах длины и в процентах. Например:

```
p {background-image : url (myimage.gif); background-position  
: 10% 1cm; color: yellow}
```

Что касается ключевых слов, то они не могут использоваться совместно с процентным значением или значением в единицах длины.

Как и для границ, отступов и полей, для фона в CSS2 предусмотрено стенографическое (сокращенное) свойство `background`, с помощью которого, можно сразу задать все параметры фона и фонового изображения. Свойство `background` объединяет в себе все пять вышеописанных свойств фона: `background-color`, `background-image`, `background-repeat`, `background-attachment` и `background-position`. Пример использования:

```
pRE ( color: blue;  
background: url (/image/bg.igf) 0%0% repeat scroll}
```

В том случае, если какой-то из параметров не указан, он будет применен в значении, используемом браузером по умолчанию. Все свойства, начинающиеся со слова `background`, не наследуются. Однако значением свойства `background-color` по умолчанию является ключевое слово `transparent`. Это означает, что фон является прозрачным и поэтому будет виден фон родительского элемента, что создает иллюзию наследования.

## разметка текста

Итак, со стилевыми блоками и их цветовыми и фоновыми характеристиками разобрались. Теперь остается описать разметку текста, располагающегося в информативной области блоков. С помощью средств языка CSS2 можно задать отступы и высоту строк в текстовых фрагментах, расстояния между словами и буквами. Возможности CSS2 также позволяют трансформировать текст, написанный строчными буквами в прописной текст, и наоборот. С помощью свойства `text-decoration` можно указать подчеркивание, зачеркивание или надчеркивание текста. Описание всего этого букета возможностей и составляет данный раздел книги.

## Отступы

С помощью свойства text-indent может быть задан отступ первого слова в первой

```
<html>

<head>

<title>
```

строке абзаца. Или, говоря другими словами, свойство text-indent определяется размер красной строки. Сдвиг производится относительно левого края текстового фрагмента (т.е. относительно левого края информативной области стилевого блока). размер отступа может быть задан либо в фиксированном виде (в единицах длины), либо в виде процентов. Процентное соотношение задается относительно ширины информативной области (ширины текстового фрагмента). Пример использования (рис. 5.22):

Демонстрация применения свойства text-indent.

```
</title>

<style type="text/css"> p (

margin:25; margin-top:5; padding:15;

background:lightgrey; border:solid gray

}

</style>

</head>

<body bgcolor=white>

<h3 align=center>Демонстрация применения свойства text-indent</h3> Обычный абзац. По умолчанию
<B>text-indent:0</B>

<p>

текст текст текст текст текст текст текст текст текст текст текст текст текст текст

</p>

Абзац, для которого указано <B>text-indent:10</B>
```



&lt;/html&gt;

p {color: yellow; text-indent: 4 em; text-align:

justify}

## Оформление текста

С помощью свойства text-decoration можно придать тексту такие визуальные характеристики, как, подчеркивание, зачеркивание и надчеркивание. Для этого достаточно использовать свойство text-decoration с одним из следующих значений:

underline каждая строка текста будет подчеркнута;

overline все строки текста будут отображены с чертой, расположенной над каждой из них;

line-through каждая строка будет отображена перечеркнутой;

по декоративного оформления текста производиться не будет. Это значение выставлено по умолчанию.

Примечание. В спецификации языка CSS2 определено также значение blink для свойства text-decoration, которое устанавливает мигание текста. Однако возможность его использования пока еще не поддерживается ни одним браузером.

```
<html>

<head>

<title>

Демонстрация применения свойства text-decoration

</title>

<style>

body {text-align:center ; background:white} </style> <body>

<h3>Демонстрация применения свойства text-decoration </h3> <hr size=20 color=lightgrey> <p>
Обычная строка текста </p>

<p style= "text-decoration:underline"> Подчеркнутая строка текста

</p>

<p style= "text-decoration:overline"> Надчеркнутая строка текста

</p>

<p style= "text-decoration:line-through"> Перечеркнутая строка текста

</p>

<hr size=20 color=lightgrey>

</body>

</html>
```

## Внутритекстовые интервалы

Важными визуальными характеристиками текста являются его межбуквенные интервалы и интервалы между словами. Управление размером межбуквенных интервалов

осуществляется с помощью CSS-свойства `letter-spacing`, которое устанавливает расстояние между буквами, задаваемое в единицах длины, например,

```
h3 {color: blue; letter-spacing: 13px}
```

Кроме того, это свойство пропорционально изменяет и интервалы между словами.

Примечание. CSS-свойство `word-spacing`, специально отвечающее за интервалы между словами, на данный момент браузерами не поддерживается. Поэтому нельзя возможностями CSS2 изменить расстояния между словами в тексте, не изменив межбуквенные расстояния.

Свойство `letter-spacing` может быть выставлено в значении `normal` (`letter-spacing: normal`), что указывает браузеру использовать стандартные интервалы текущего шрифта. размер междустрочных интервалов задается с помощью свойства `line-height`.



## бота с регистром

С помощью свойства `text-transform`, определенного в рамках спецификации CSS2, можно осуществлять преобразование регистра букв текста. Или, другими словами, трансформировать строчные буквы в прописные, и наоборот. Для этих целей свойство `text-transform` может принимать следующие значения:

`capitalize` первая буква каждого слова отображается прописной (заглавной);

`uppercase` . все буквы каждого слова отображаются прописными (заглавными);

`lowercase` . все буквы каждого слова отображаются как строчные;

`no` . никакого преобразования регистра не производится.

Пример использования:

```
H1 {color: green; letter-spacing: 5px;
```

text-transform: uppercase}

# Задание шрифта

## *средствами CSS2*

От выбора подходящего шрифта может во многом зависеть привлекательность документа. Каждый из них имеет свой индивидуальный внешний вид. Не буду останавливаться на особенностях различных шрифтов, это выходит за рамки данной книги. В этом разделе будет произведено описание работы со шрифтами. Подключение определенного шрифта и задание его характеристик (прямой, курсив, жирный и т.п.) является одними из наиболее часто производимых разработчиком операций. Язык каскадных таблиц стилей CSS2 предлагает для этих целей свои возможности.

### *Подключение шрифтов. Свойство font-family*

С помощью свойства font-family браузеру указывается шрифт, которым он должен отображать текстовую информацию, для которой это свойство определено (или наследуется). Например:

```
p { font-family:"Arial" }
```

В результате такого задания текст абзацев будет отображаться шрифтом Arial.

Возможности свойства font-family таковы, что оно позволяет указывать приоритетный список названий шрифтов, каждый из которых будет пробовать подключаться в соответствии со своим приоритетом. Например, правило

body ( font-family : "Heretet", "Times New Roman", "Arial" ) означает, что сначала браузер будет пробовать использовать шрифт Heretet. Если такой шрифт отсутствует в его распоряжении (его просто нет на компьютере пользователя), то будет произведена попытка использования шрифта Times New Roman и т.д.

К тому же, для свойства font-family можно заказывать целое семейство шрифтов, среди которых браузер будет искать шрифт, подходящий для отображения всех символов текста. Возможность такого задания введена в язык CSS2 с целью максимально корректной обработки наихудшего варианта, когда браузеру не удастся использовать ни один из специально установленных разработчиком шрифтов.

Ниже приведены пять типовых семейств шрифтов, включающие в себя все наиболее распространенные шрифты, установленные практически на всех компьютерах:

- Serif (например, Times New Roman);
- Sans-serif (например, Helvetica);
- Monospace (например, Courier New);
- Curcive (например, Zapf-Chancery);
- Fantasy (например, Western).

Полный список шрифтов с указанием их принадлежности к различным группам приведен в разделе "Подключение удаленных шрифтов". Пример использования:

```
body{ font-family : "Times New Roman", "Times", serif}
```

## *Указание стиля шрифта. Свойство font-style*

С помощью свойства font-style можно задавать стиль шрифта. рассмотрим значения, принимаемые свойством font-style и соответствующий им эффект:

normal — шрифт отображается в своем обычном, прямом виде. Это значение используется по умолчанию;

italic — шрифт будет отображен курсивом;

oblique — шрифт будет отображен в наклонном виде.

Примечание. По идее, начертание шрифта в значении italic должно отличаться от начертания шрифта в значении oblique. Однако для браузера Internet Explorer разницы между italic и oblique никакой нет, а браузер Netscape Communicator вообще игнорирует свойство font-style в значении oblique. Поэтому для наклонного (курсивного) изображения текста используйте значение italic.

размер шрифта. Свойство font-size

В качестве значения свойства font-size могут выступать либо ключевые слова, либо единицы длины, характеризующие высоту шрифта (при этом ширина изменяется пропорционально высоте).

Возможно также процентное задание размера шрифта по отношению к размеру шрифта, используемому по умолчанию. Например, правила B {font-size : 150%} и B {font-size : 1.5 em} являются абсолютно эквивалентными.

Для свойства font-size определена следующая последовательность ключевых слов:

xx-small, x-small, small, medium, large, x-large, xx-large

## *Свойство font-variant*

В рамках спецификации CSS2 определено такое экзотическое свойство, как font-variant. Им задается стиль шрифта, при котором все буквы текста будут выглядеть как прописные (заглавные), но немного меньшего размера, чем обычные прописные буквы. На текущий момент времени свойство font-variant не поддерживается браузером Netscape Communicator. Свойство font-variant является наследуемым и может принимать следующие значения:

small-caps при этом происходят вышеописанные изменения шрифта;

normal при таком значении свойства никаких изменений не происходит.

Используется по умолчанию.

## Жирность текста. Свойство font-weight

Жирность шрифта, которым отображается текст, может быть задана через свойство font-weight. В спецификации CSS2 регламентируется использование 9 градаций жирности для одного шрифта. Однако это еще не означает, что сам шрифт все эти девять градаций поддерживает. Например, используемый по умолчанию большинством браузеров шрифт Times New Roman поддерживает только четыре градации жирности. Каждая из этих девяти градаций задается одним числом: 100, 200, 300.....900. Значение 100 соответствует самому бледному

рис. 5.25. Демонстрация применения свойства font-weight для шрифта Times New Roman

варианту шрифта, значение 900 — самому жирному. Кроме этого, допустимо использование следующих ключевых слов:

normal задает нормальную (обычную) жирность шрифта. Это значение ис

пользуется по умолчанию;

bold указывает на использование полужирного варианта шрифта.

Ключевое слово normal соответствует числовому значению 400, слово bold — значению 700. Как уже писалось выше, для разных шрифтов может существовать разное количество градаций жирности, и разное числовое соответствие для них. Для шрифта Times New Roman имеет место следующие значения (рис. 5.25):

```
<html>

<head>

<title>

Демонстрация применения свойства font-weight для шрифта Times New Roman

</title>

<style>

body {text-align:center,• background:white) </style> <body>

<hr size=10 color=lightgrey>

<h3>Демонстрация применения свойства font -weight</h3> <p style= " font-weight:100">

Строка текста, для которой установлена жирность 100 </p>

<p style= "font-weight:200">
```

Строка текста, для которой установлена жирность 200 </p>

<p style= "font-weight:300">

Строка текста, для которой установлена жирность 300 </p>

<p style= "font-weight:400">

Строка текста, для которой установлена жирность 400 </p>

<p style= "font-weight:500">

Строка текста, для которой установлена жирность 500 </p>

<p style= "font-weight:600">

Строка текста, для которой установлена жирность 600 </p>

<p style= "font-weight:700">

Строка текста, для которой установлена жирность 700

<p style= "font-weight:800">

Строка текста, для которой установлена жирность 800

</p>

<p style= "font-weight:900">

Строка текста, для которой установлена жирность 900

</p>

<hr size=10 color=lightgrey>

</body>

</html>

Этим правилом для текстового содержимого элемента body установлен полужирный курсивный шрифт Times New Roman размером 12 пунктов при высоте строки 14 пунктов. В том случае, если шрифт Times New Roman будет не найден браузером, задано использование любого шрифта из семейства serif. Другие примеры:

```
p{font: 900 12pt/14pt serif}
```

```
div {font: normal small-caps 120%/1.2 em "Helvetica"} pRE {font: italic 600 12pt;
```

```
color: green}
```

Подключение удаленных шрифтов. Правило @font-face

Эй-ти правило @font-face позволяет разработчику задавать свои, авторские шрифты. При этом если эти шрифты отсутствуют на клиентском компьютере, то они подгружаются с сервера, URL-адрес которого задается одним из параметров Эй-Ти правила @font-face. Пример использования:

@font-face (font-family: My Font;

src: url (<http://myserver.da.ru/Font/MyFont.eot>))}

Параметром font-family указывается имя шрифта, под которым он будет использоваться для вывода текста. Параметр src указывает браузеру URL-адрес, где его можно взять. После такого описания шрифт My Font готов к полноценному использованию. Например:

```
<style>

@font-face { font-family: My Font;

src: url[ (http://myserver.da.ru/Font/MyFont.eot)](http://myserver.da.ru/Font/MyFont.eot)) p
(color: yellow;

font-family: "MyFont"; font-size: 12ptt

</style>
```

Ниже приведен примерный список принадлежности шрифтов к различным семействам.

## *СЕМЕЙСТВО SERIF*

Латинские шрифты: Times New Roman, Bodoni, Garamond, Minion Web, ITC Stone Serif, MS Georgia, Bitstream Cyberbit;

Греческие шрифты: Bitstream Cyberbit;

Кириллические шрифты: Adobe Minion Cyrillic, Excelcior Cyrillic Upright, Monotype Albion 70, Bitstream Cyberbit, ER Bukinst;

■ ■ Еврейские шрифты: New peninim, Raanana, Bitstream Cyberbit; + Японские шрифты: Ryumin Light-KL, Kyokasho ICA, Futo Min A101; + Арабские шрифты: Bitstream Cyberbit.

## *СЕМЕЙСТВО SANS-SERIF*

Латинские шрифты: MS Trebuchet, ITC Avant Garde Gothic, MS Arial, MS Verdana, Univers, Futura, ITC Stone Sans, Gill Sans, Akzidenz Grotesk, Helvetica;

Греческие шрифты: Attika, Typiko New Era, MS Tahoma, Monotype Gill Sans 571, Helvetica Greek;

■ Кириллические шрифты: Helvetica Cyrillic, ER Univers, Lucida Sans Unicode, Bastion;

Еврейские шрифты: Arial Hebrew, MS Tahoma;

Японские шрифты: Shin Go, Heisei Kaku Gothic W5;

Арабские шрифты: MS Tahoma.



## *СЕМЕЙСТВО CURSIVE*

Латинские шрифты: Caflisch Script, Adobe poetica, Sanvito, Ex ponto, Snell Roundhand, Zapf-Chancery;

Кириллические шрифты: ER Architekt;

Еврейские шрифты: Corsiva;

Арабские шрифты: DecoType Naskh, Monotype Urdu 507.

## *СЕМЕЙСТВО FANTASY*

Латинские шрифты: Alpha Geometrique, Critter, Cottonwood, FB Reactor, Studz.

## *СЕМЕЙСТВО MONOSPACE*

- Латинские шрифты: Courier, MS Courier New, prestige, Everson Mono;
- Греческие шрифты: MS Courier New, Everson Mono;
- Кириллические шрифты: ER Kurier, Everson Mono;
- Японские шрифты: Osaka Monospaced.