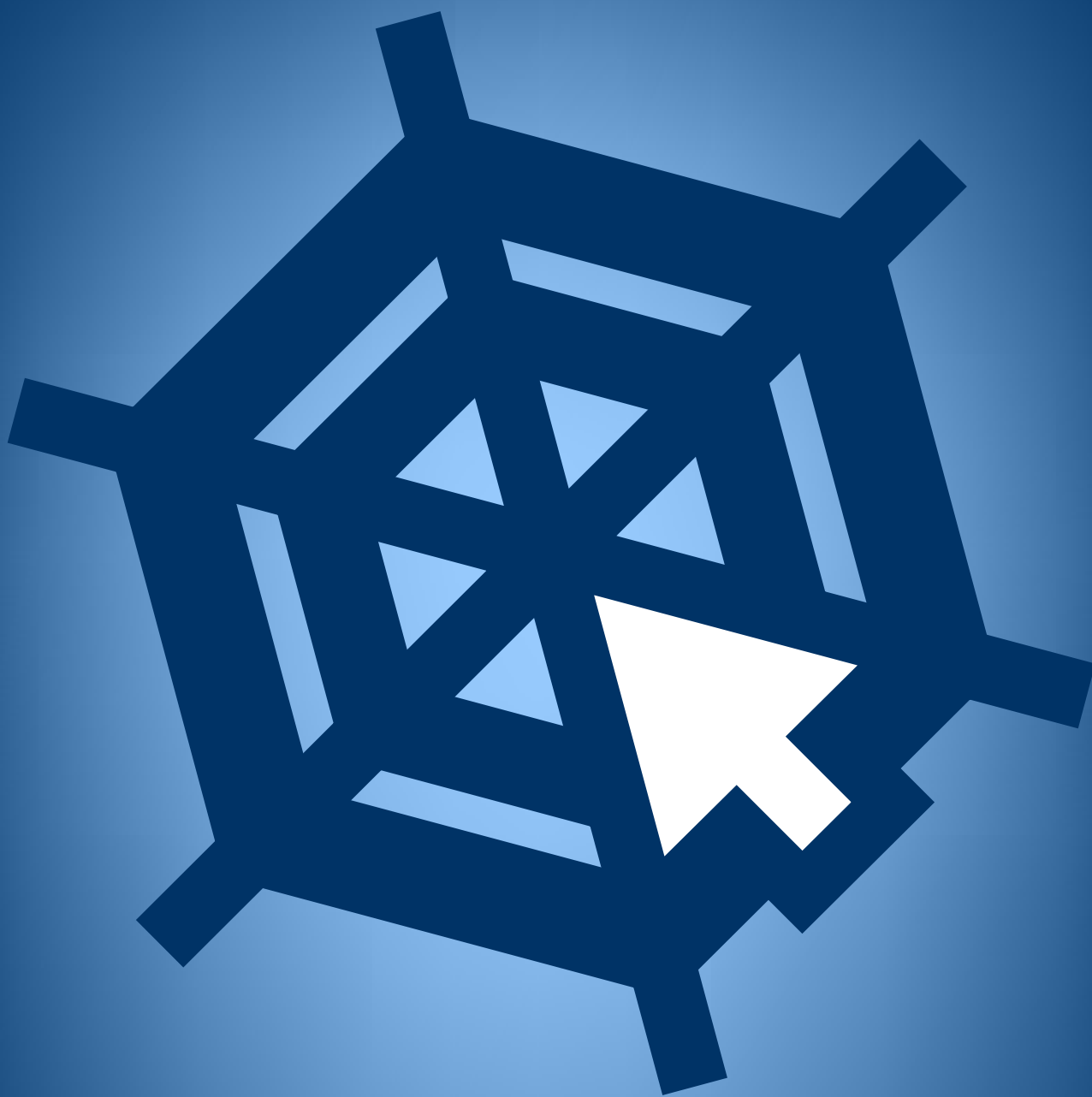


Влад Мержевич



**Приемы верстки
веб-страниц**

СОДЕРЖАНИЕ

Вступление	3
Благодарности	3
Поддержка	3
1. ОСОБЕННОСТИ ВЕРСТКИ ВЕБ-СТРАНИЦ	4
2. ИСПОЛЬЗОВАНИЕ ФОНОВОГО РИСУНКА	6
3. СОЗДАНИЕ РАМОК	9
4. ВЫРАВНИВАНИЕ ПО ЦЕНТРУ	14
5. ЦЕНТРИРОВАНИЕ ПО ВЕРТИКАЛИ	16
6. ВЫРАВНИВАНИЕ ЭЛЕМЕНТОВ	16
7. ВЕРТИКАЛЬНЫЕ ЛИНИИ	19
8. ГОРИЗОНТАЛЬНЫЕ ЛИНИИ	21
9. СОЗДАНИЕ КОЛОНОК	24
10. КОЛОНКИ С РАЗДЕЛИТЕЛЕМ	24
11. ПОДРИСУНОЧНАЯ ПОДПИСЬ	25
12. СОЗДАНИЕ БУКВИЦЫ	27
13. ОТСТУП ПЕРВОЙ СТРОКИ	29
14. ОТСТУПЫ НА ВЕБ-СТРАНИЦЕ	30

Вступление

В этой книге вы не найдете описания тегов, здесь не дается структура HTML-документа и ничего не говорится о вставке изображений и их создании. Вы не узнаете ни про параметры стилей, ни про скрипты, ни про формы, ни про фреймы. Ничего этого здесь нет. Но зато есть другое, более ценное, то, с чем сталкивался каждый начинающий веб-мастер, который изучил HTML и приступил к созданию сайтов - решение тех задач, над которыми многие ломают голову и задают себе вопрос: «как же это сделано?». Ответы на эти вопросы собраны здесь, в этой книге, в простой и удобной форме с примерами и подробным описанием.

Верстка веб-страниц – не просто знание приемов и хитростей создания различных эффектов. Это умение предугадывать результат действий с элементами веб-страниц и понимание особенностей различных браузеров, которые могут по-разному показывать сайт. Работа верстальщика сродни работе шеф-повара, который точно знает, какие ингредиенты и в каком количестве нужно добавить, чтобы улучшить вкус конечного блюда. Вооружившись простыми, но действенными рецептами верстки, которые даются в этой книге, вы сможете создавать впечатляющие сайты. Добро пожаловать на кухню верстки веб-страниц!

Благодарности

Выражаю признательность своему учителю и наставнику Петру Никифоровичу Сильченко за его неоценимые советы по написанию научных работ. Благодаря ему, стиль и удобочитаемость текстов улучшились.

Также хочу сказать спасибо Егору Москвичеву за множество полезных рекомендаций по содержанию сайта, которые он мне дал.

Поддержка

Ваши замечания, предложения и вопросы по книге направляйте автору по адресу электронной почты vlad@htmlbook.ru.

Все описанные в книге материалы и примеры можно посмотреть в Интернете на сайте автора по адресу:
<http://www.htmlbook.ru/layout>

Исправления и дополнения по книге, а также новые версии доступны по адресу:
<http://www.htmlbook.ru/book>

1. ОСОБЕННОСТИ ВЕРСТКИ ВЕБ-СТРАНИЦ

При верстке веб-сайта, несмотря на некоторую аналогию его с печатными изданиями, следует все же помнить, что мы имеем дело с компьютерным текстом, который зависит от ряда факторов. Учет этих особенностей необходим для создания удобных для восприятия и пользования материалов, которые будут выглядеть одинаково в различных браузерах.

Ширина страницы зависит от размера и разрешения монитора

Данная особенность породила метод верстки веб-страниц, основанный на использовании таблиц с невидимой границей. Идеология HTML рекомендует использовать таблицы только по их прямому назначению, т.е. для создания именно таблиц. Но что делать, располагая столь слабыми средствами верстки, приходится использовать разные ухищрения.

Все многообразие дизайна можно свести к двум типам: «фиксированный» и «резиновый» дизайн.

При «фиксированном» дизайне макет страницы создается на основе таблиц или слоев определенной ширины. Выбор ширины зависит от ориентации на определенное разрешение монитора пользователей. Так, для разрешения 640×480 пикселей размер таблицы следует брать не более 619 пикселей. Для всех пользователей сайт, построенный на таком макете, будет отображаться корректно, но при больших разрешениях оказывается слишком много пустого места. Поскольку наиболее популярным среди пользователей Интернета является разрешение 800×600, разработчики сайтов ориентируются именно на него и делают таблицы шириной 700-770 пикселей.

Для «резинового дизайна» характерно использование таблиц с шириной, заданной в процентном отношении от ширины браузера. Достоинства – используется все доступное пространство веб-страницы, недостаток – при большом разрешении монитора текст сложно читать. Попробуйте, почитайте страницу со строчками длиной символов 200, глаза устанут по ней бегать.

Хотя и «резиновый» и «фиксированный» дизайн можно увидеть на различных сайтах повсеместно, в последнее время предпочтение все больше отдается гибкой структуре.

Размер шрифта зависит от настроек операционной системы и браузера

Не любят разработчики сайтов, когда их пользователи изменяют размер шрифта. Потому что в этот момент могут проявляться всякие разные недочеты верстки. И чтобы пользователь не мог менять размер шрифта, эта величина жестко прописывается в стилях. Даже если текст мелкий-мелкий, изменить его простыми средствами не представляется возможным. Поэтому лучше задавать размер шрифта в процентах, например: *font-size: 150%* Тогда в опциях браузера размер шрифта можно делать большим или меньшим, что невозможно при абсолютном задании размера шрифта в пунктах. Это позволит установить размер шрифта подходящим для комфортного чтения. Только надо проверить вначале, что все на сайте корректно работает при любом размере шрифта.

Несколько идущих подряд пробелов показываются как один

Сколько бы пробелов не стояло между словами, все равно на веб-странице мы увидим лишь один. В общем, правильное решение, чего лишний раз пробелы плодить. Но как же тогда быть с красными строками и другими отступами, которые так иногда необходимо использовать. Вот тут и полюбился многим символ неразрывного пробела (` `) за то, что позволяет делать столько подряд идущих пробелов, сколько хочется.

Нет переносов слов в тексте

К сожалению, HTML не поддерживает перенос слов, так как это делают текстовые редакторы. Это условие совершенно несущественно, пока не используется выравнивание текста по ширине. В этом случае, блок текста выравнивается по левому и правому краю. Короткие строки при этом растягиваются за счет автоматического добавления пробелов между словами. Иногда пустые блоки между словами настолько велики, что портят внешний вид страницы и ухудшают читабельность текста.

Количество используемых для текста шрифтов ограничено

Разумеется, можно указать любой шрифт, установленный у вас в операционной системе и он будет корректно показываться на данном компьютере. Но никто при этом не гарантирует, что точно такой же шрифт будет находиться на удаленном компьютере пользователя. Поэтому для совместимости используется всего несколько шрифтов, которые встречаются повсеместно: Arial, Times, Verdana, Courier, Comic Sans MS. В случае если хочется воспользоваться понравившимся шрифтом, следует создать картинку в графическом редакторе, содержащую надпись.

Использование изображений вместо текста

Особенность непосредственно связана с предыдущим пунктом. Количество шрифтов, которые можно применять на веб-странице ограничено, а стили неудобно или невозможно применять для управления некоторыми параметрами текста, как это позволяют делать текстовые или графические редакторы. Поэтому некоторые блоки текста, например, заголовки и меню вставляют в виде изображения.

Высота содержимого страницы не ограничена рамками окна браузера

Несмотря на очевидность данного воззрения, из него следуют некоторые выводы:

- поскольку высота текста на странице не ограничена определенными рамками, как на листе бумаги и напрямую зависит от разрешения монитора, размера и типа шрифта и еще от кучи других параметров, вычислить ее не представляется возможным;
- вертикальные линии привязанные к высоте текста можно строить только с использованием таблиц.

Использование распорок

Распоркой называется прозрачный рисунок, как правило, размером 1 на 1 пиксел, предназначенный для контроля пустого пространства. Распорки

вставляются в ячейки таблиц, чтобы они не меняли свой размер при изменении окна браузера или разрешения монитора.

Написание программных скриптов вместо кода HTML

HTML не является языком программирования и имеет массу ограничений. Чтобы изменить его статическую природу, для создания различных эффектов используются скрипты – программы, которые встраиваются в тело веб-страницы и исполняются на стороне клиента. С помощью скриптов можно создавать на сайте игры, различные меню, динамически управлять стилями и многое другое.

2. ИСПОЛЬЗОВАНИЕ ФОНОВОГО РИСУНКА

Используя фоновый рисунок веб-страницы или таблицы можно создавать различные графические эффекты на сайте, например, вертикальные и горизонтальные линии, тени, полосы и многое другое.

Фон веб-страницы

Чтобы на веб-странице с левого края шла вертикальная фоновая полоса, используется следующий прием. Создаем рисунок шириной примерно 2000 пикселей. Такая большая ширина нужна для того, чтобы на всех мониторах независимо от их разрешения страница выглядела бы одинаково (рис. 2.1). С левой части рисунок зарисовывается нужным цветом желаемой ширины. Можно создавать не только прямоугольник, но и любые декоративные изображения, главное, чтобы они состыковывались по вертикали. Полученный рисунок вставляем на веб-страницу как обычный фон параметром *background* тега BODY. Поскольку рисунок широк и не помещается на всю ширину экрана монитора, на странице он будет повторяться только вниз.

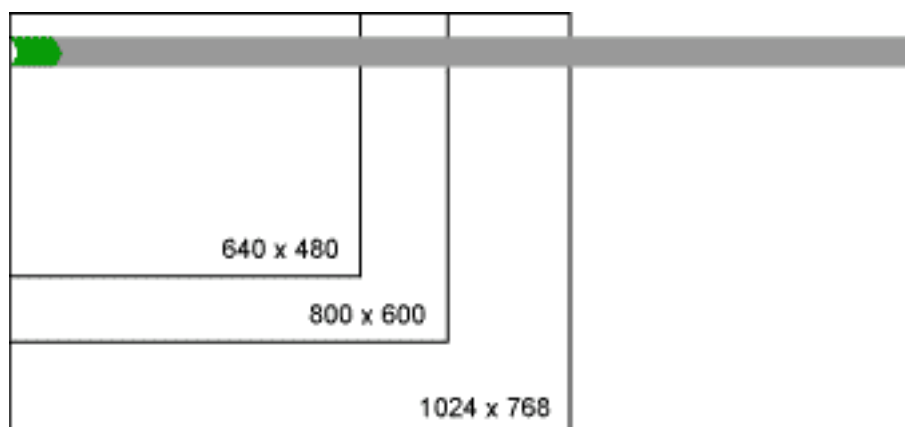


Рис. 2.1. Фоновый рисунок для различных разрешений монитора

В итоге получим страницу, у которой по левому краю идет декоративная полоса (рис. 2.2).



Рис. 2.2. Результат использования широкого фона

Приведенный прием можно использовать для создания и горизонтальных полос. В этом случае, фоновое изображение должно располагаться вертикально.

Для размещения содержимого страницы строго в определенных областях, можно использовать таблицу с нулевой границей как каркас (рис. 2.3). Самой таблицы видно не будет, но она позволяет отделить одну часть веб-страницы от другой, например, ссылки от содержания.



Рис. 2.3. Использование таблицы для разделения материала

Аналогичный подход можно использовать и с таблицами. Только вначале создается таблица, а уже в одной из ее ячеек размещается фоновый рисунок. Ширина ячеек задается фиксированной и зависит от фонового рисунка, а чтобы она не менялась, внутри ячейки рекомендуется поместить распорку – прозрачный рисунок заданных размеров (рис. 2.4).



Рис. 2.4. Фоновый рисунок для ячейки таблицы

Код для данного использования таблицы с фоном приведен в Примере 2.1.

Пример 2.1. Создание фона с помощью таблицы

```
<table width="300" cellspacing="0" cellpadding="0">
<tr>
  <td width="82" background="bg.gif">Меню</td>
  <td width="200" bgcolor="#CCCCCC">Текст</td>
</tr>
</table>
```

Рисунок *bg.gif* является фоном, а *spacer.gif* служит распоркой. В правой ячейке фон залит серым цветом, что можно делать по желанию.

Создание вертикальной линии

Очень удобно и просто делать вертикальные линии с помощью таблицы и фонового рисунка. Вначале готовим изображение в графическом редакторе, это может быть сплошная, пунктирная или любая декоративная линия (рис. 2.5).

Почему рисунок именно такого размера и формы описано чуть ниже.



Рис. 2.5. Заготовка для создания линии

Следующий этап – создание таблицы. Ширина одной из ячеек должна совпадать с толщиной линии, обычно это 1-2 пиксела. Внутри нее и размещаем фоном изображение.

Пример 2.2. Создание вертикальной линии

```
<table width="300" border="0" cellspacing="0" cellpadding="0">
<tr>
  <td width="2" background="line.gif"></td>
  <td width="10"></td>
  <td width="288"><i>При истечении возможности понимания
вышеизложенной информации вы имеете объективную возможность подать
официальный запрос главному субординатору локальной виртуальной
вселенной.</i>
</td>
</tr>
</table>
```

Если ширина таблицы фиксирована, лучше задавать точные размеры всех ячеек. При ширине заданной в процентах, возможны небольшие изменения размеров ячеек. Вот для такого случая и делается изображение линии (рис. 2.5) с пустым полем справа. Если этого не сделать, линия может получиться несколько толще, чем ожидалось. А так, если ширина ячейки и будет увеличиваться, несмотря на прописанные размеры, вид линии останется неизменным. Вдобавок, такой «запас» позволяет изменять значение параметра *cellpadding*, который управляет отступом от края таблицы до ее содержимого, без отрицательных последствий.

Рисунок достаточно делать шириной 20-30 пикселей, а цвет фона рисунка должен быть таким же, как у веб-страницы.

3. СОЗДАНИЕ РАМОК

Информация обведенная рамкой позволяет отделить один материал на веб-странице от другого и привлечь внимание читателя. Кроме того, рамка вносит элегантность в оформление сайта.

Для создания рамок применяется несколько способов, их которых большая часть использует свойства таблиц. Стили дополняют возможности таблиц и позволяют сделать рамки разных видов простыми средствами. Ниже приводятся распространенные приемы получения рамок, которые можно встретить повсеместно на разных сайтах.

Использование параметра таблицы `border`

Самый простой способ создания рамки заключается в использовании параметров таблицы `border`, определяющего толщину рамки и `bordercolor`, который задает ее цвет. Рамки созданные таким образом отличаются по своему виду в разных браузерах (рис. 3.1).

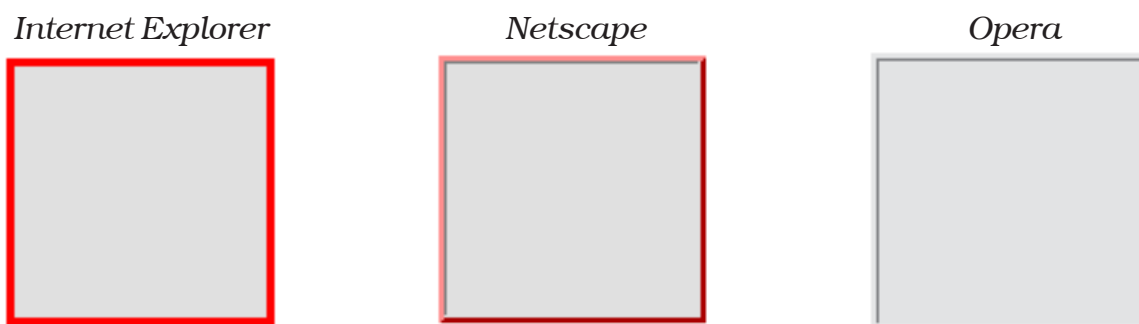


Рис. 3.1. Рамка полученная с помощью параметра `border`

Netscape имитирует трехмерность рамки, браузер Opera вообще оставляет ее цвет без изменения, а Internet Explorer делает рамку сплошной.

Пример 3.1. Использование параметра `border`

```
<table border="2" bordercolor="#FF0000" width="100" height="100"
cellpadding="6" cellspacing="0" bgcolor="#E0E0E0">
<tr>
<td>Содержимое</td>
</tr>
</table>
```

Параметр `border` определяет толщину рамки, `bordercolor` – ее цвет, `cellpadding` задает расстояние от рамки до содержимого таблицы.

Вложенные таблицы

Еще один простой и универсальный способ построения рамки, работающий одинаково в разных браузерах, основан на наложении двух таблиц друг на друга. Если мы возьмем прямоугольник из, например, красной бумаги и сверху на него

наложим еще один прямоугольник белого цвета, чуть меньшего размера, то увидим красную рамку. Только вместо листа бумаги используем тег TABLE, а цвет таблицы задаем параметром *bgcolor*:

Пример 3.2. Использование вложенных таблиц

```
<table width="300" height="300" border="0" cellspacing="0"
cellpadding="1" bgcolor="#FF0000">
<tr>
<td>
<table border="0" cellspacing="0" cellpadding="10" bgcolor="#FFFFFF"
width="300" height="300">
<tr><td align="center">Содержимое</td></tr>
</table>
</td></tr>
</table>
```

В верхней таблице устанавливаем ширину и высоту таблицы по желанию, параметром *bgcolor* задаем цвет рамки, *cellspacing* присваиваем нулю, а *cellpadding* управляет толщиной границы. Чем этот параметр больше, тем толще будет и рамка.

Для внутренней таблицы надо обязательно задать ее цвет, отличный от цвета внешней, совпадающий с цветом фона веб-страницы, в примере, например, задан белый. Параметр *cellspacing* или *cellpadding* (в данном случае без разницы какой использовать) определяет расстояние от границы рамки до содержимого таблицы.

Ширина и высота внутренней таблицы обязательно должны совпадать с шириной и высотой внешней таблицы.

Замечание: Перенос строки в тексте кода HTML автоматически добавляет пробел. Если рамка в некоторых местах получается толще, чем задумано, следует убрать лишние пробелы между тегами и тем самым сократить количество строк.

Заливка ячеек таблицы цветом

Использование цвета фона ячеек таблицы – один из универсальных и распространенных способов создания рамки. Создаем таблицу (рис. 3.2) и заливаем крайние ячейки нужным цветом (рис. 3.3).



Рис. 3.2



Рис. 3.3

Толщина рамки определяется шириной и высотой ячеек. Обязательно следует поместить внутри этих ячеек таблицы распорку – прозрачный рисунок размером 1 на 1 пиксел (в примере он называется *spacer.gif*), иначе в браузере Netscape этот способ работать не будет.

Пример 3.3. Создание рамки заливкой цветом ячеек таблицы

```
<table width="200" border="0" height="200" cellspacing="0"
cellpadding="0">
<tr bgcolor="#FF6633">
<td width="2" height="2">
</td>
<td height="2"></td>
<td width="2" height="2"></td></tr>
<tr><td width="2" bgcolor="#FF6633"></td>
<td align="center">Содержимое</td>
<td width="2" bgcolor="#FF6633"></td></tr>
<tr bgcolor="#FF6633">
<td width="2" height="2"></td>
<td height="2"></td>
<td width="2" height="2"></td></tr>
</table>
```

В данном примере делается красная рамка толщиной 2 пиксела, она смотрится наиболее элегантно. Параметры таблицы *cellspacing* и *cellpadding* приравниваются нулю, чтобы рамка оставалась нужной толщины и без зазоров. Если поместить внутрь такой таблицы текст, он будет плотно прилегать к рамке, что некрасиво и плохо читается. Чтобы добавить отступы можно создать для этой цели дополнительные ячейки, использовать вложенную таблицу или воспользоваться стилями.

Замечание: Приведенный способ чувствителен к разным параметрам, поэтому, в случае неправильной отрисовки рамки следует проверить следующее:

- если размер таблицы задается в пикселах, надо задать фиксированные размеры всех ячеек;
- когда ширина таблицы указана в процентах, внутри рамки нужно поместить текст, который «распирает» внутреннюю ячейку до нужной ширины; при отсутствии текста или другого содержимого внутри таблицы с рамкой, ее ширина окажется гораздо шире желаемой;
- в ячейках, где проходит рамка, не должно быть никаких символов неразделяемого пробела (* *), что так любят автоматически добавлять некоторые редакторы веб-страниц;
- перенос строки автоматически добавляет в текст пробел, поэтому между открывающим и закрывающим тегами `<TD>` и `</TD>` текст следует писать в одну строку, чтобы не было зазоров между ячейками.

Декоративная рамка

Для создания желаемой рамки ее необходимо предварительно нарисовать в каком-нибудь графическом редакторе. На рис. 3.4, например, показана рамка со скругленными углами. Затем это изображение разрезаем на 9 частей, помеченные на рисунке 3.5 оранжевыми линиями и номерами. Разрезать изображение можно, например, в программе ImageReady или другой подобной.



Рис. 3.4

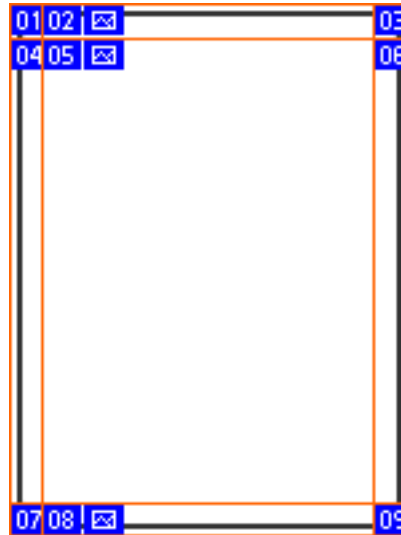






Рис. 3.5

После разрезания рисунка на части получим 8 фрагментов:

Рисунок	Положение	Имя файла
	Левый верхний угол	1.gif
	Верхняя горизонтальная линия	2.gif
	Правый верхний угол	3.gif
	Левая вертикальная линия	4.gif
	Правая вертикальная линия	6.gif
	Левый нижний угол	7.gif
	Нижняя горизонтальная линия	8.gif
	Правый нижний угол	9.gif

Пятого фрагмента рамки не будет (5.gif), поскольку у нас его заменит содержимое. Теперь создаем таблицу размером 3x3, как показано на рис. 3.2 и в ее ячейках размещаем созданные рисунки.

Пример 3.4. Создание декоративной рамки

```
<table width="400" border="0" cellspacing="0" cellpadding="0">
<tr>
<td height="13"></td>
<td background="2.gif" height="13"></td>
<td height="13"></td>
</tr>
<tr>
<td background="4.gif" width="12">&nbsp;</td>
<td align="center">Содержимое</td>
<td background="6.gif" width="12">&nbsp;</td>
</tr>
<tr>
<td height="13"></td>
<td background="8.gif"></td>
<td height="13"></td>
</tr>
</table>
```

В таблице параметры *border*, *cellspacing* и *cellpadding* должны быть равны нулю, иначе линии не будут состыковываться между собой.

Уголки вставляем как обычные рисунки с помощью тега IMG, а вот горизонтальные и вертикальные линии следует задавать с помощью фона ячейки таблицы, параметром *background*. Это позволит сделать нашу рамку масштабируемой и размер ее будет изменяться в зависимости от содержимого.

Чтобы линии плотно прилегали друг к другу и между ними не было пробелов надо обязательно указывать все размеры ячеек и рисунков.

В ячейках, где фоном размещаются горизонтальные и вертикальные линии, необходимо поместить распорку – это либо пробел, либо прозрачный рисунок формата GIF размером 1 на 1 пиксел (в примере файл называется *spacer.gif*). Делается это для того, чтобы обмануть браузер Netscape, который не показывает содержимое ячейки, если она пуста.

Приведенный способ позволяет создавать рамки практически любого вида.

Использование стилей

С помощью стилей рамку можно применить к любому блочному тегу, например, параграфу (тег P), таблице и тегу DIV. Стили позволяют создать рамку проще и удобней, чем с использованием таблиц и предоставляют разные виды рамок, которые показаны на рис. 3.6.

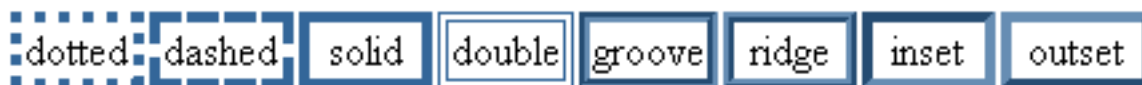


Рис. 3.6 Вид рамок задаваемых с помощью стилей

Первые две типа рамок – *dotted* и *dashed* поддерживаются браузерами Netscape и Internet Explorer только со старших версий.

Рамка задается проще всего параметром *border*, который сразу указывает ее вид, толщину и цвет.

Пример 3.6. Использование стилей

```
<html>
<head>
<style type="text/css">
P { border: double 4px #336699; padding: 5px }
</style>
</head>

<body>
<p>При работе на вычислительной технике необходимо сесть так, чтобы
руки с предплечьями образовывали прямой угол, глаза поставить на
расстояние 30-40 см от рабочей поверхности монитора.
</p>
</body>
</html>
```

В примере используется двойная рамка, толщина которой складывается из толщины линий и расстояния между ними. Чтобы рамка не соприкасалась с текстом, задан отступ от рамки до содержимого параметром *padding*.

Замечание: Браузер Netscape 4.x не добавляет рамку для таблиц и ограничивает ширину рамки ее содержимым.

4. ВЫРАВНИВАНИЕ ПО ЦЕНТРУ

Казалось бы, нет ничего сложного, чтобы установить картинку или текст в центре окна браузера. Используем тег CENTER и все становится на то место, куда мы и задумывали. Однако не все так просто. Существует три способа выравнивания по центру, каждый из них имеет свои особенности и отличия в разных браузерах.

Один из самых простых и удобных тегов – CENTER предназначен для выравнивания блока текста. С помощью этого тега также можно центрировать рисунки и таблицы. Исключение составляют элементы `` или `` для которых выравнивание задается свойствами тега IMG. Так, если поместить элемент `` внутри тега CENTER, рисунок окажется выровненным по правому краю.

Пример 4.1. Центрирование с помощью тега CENTER

```
<center>
Данный текст будет выровнен по центру окна браузера, а нижележащий
рисунок по правому краю.
<img src=no.gif width=100 height=100 align=right>
</center>
```

Формально CENTER должен использоваться только как параметр других блочных тегов (P, PRE и другие). Однако в браузере Netscape Navigator 2.0 CENTER был введен как самостоятельный тег. Это дополнение было призвано убрать лишний вертикальный отступ, который появляется при использовании блочных тегов. Если вместо тега CENTER текст поместить внутрь параграфа (`<p align="center">`), появляется дополнительный вертикальный отступ между горизонтальной линией и этим текстом.

CENTER не входит в спецификацию HTML. В «официальном» HTML форматирование, такое как выравнивание текста должно производиться через атрибуты тегов (например, `<p align="center">`) или с использованием стилей. Тем не менее, этот тег обрел право на существование. Однако после опубликования спецификации HTML 4 W3 Консорциум рекомендовал воздержаться от использования тега CENTER, а вместо него использовать элемент `<div align="center">... </div>`, как показано в примере 4.2.

Пример 4.2. Центрирование с помощью тега DIV

```
<hr>
<div align="center">
Выравнивание текста по центру с помощью тега DIV
</div>
<hr>
```

Еще один способ центрирования заключается в использовании стилей. Стили представляют собой инструкции, которые позволяют управлять атрибутами форматирования как шрифт, цвет, выравнивание и др. Пример переопределения тега P для центрирования текста.

Пример 4.3. Центрирование с помощью стилей

```
<html>
<head>
<style type="text/css">
P { text-align: center; }
</style>
</head>

<body>
<p>Теперь, когда вы будете использовать тег P с вышеуказанным
стилем, текст параграфа будет выровнен по центру окна браузера </p>
</body>
</html>
```

Для установки текста по центру используется параметр `text-align: center`.

5. ЦЕНТРИРОВАНИЕ ПО ВЕРТИКАЛИ

Одним из способов показать посетителю направленность и название сайта является использование сплэш-страницы. Это первая страница, на которой, как правило, расположен рисунок или flash-заставка, выражающие главную идею сайта. Это изображение также является и ссылкой на остальные разделы сайта. Задача – разместить такой рисунок строго по центру окна браузера вне зависимости от разрешения монитора. Добавление нескольких тегов BR перед рисунком является неверным, поскольку при изменении разрешения или размера шрифта изменится и положение изображения. Правильное решение следующее. Необходимо поместить рисунок, центрированный по горизонтали, внутрь таблицы со 100% шириной и высотой, как показано в Примере 5.1.

Пример 5.1. Центрирование рисунка по вертикали

```
<table width="100%" height="100%">
<tr>
<td align="center">

</td>
</tr>
</table>
```

По вертикали ячейку можно не центрировать, это положение задается по умолчанию, а по горизонтали используется параметр *align="center"* тега TD.

6. ВЫРАВНИВАНИЕ ЭЛЕМЕНТОВ

Выравнивая различные элементы веб-страницы по невидимой линии, мы тем самым зрительно связываем их друг с другом. Такие элементы, как текст и графические изображения с помощью выравнивания можно объединить в одну группу, показывая их связь между собой.

Выравнивание по горизонтали

Задача следующая, нужно выровнять два изображения, находящихся на одной горизонтальной линии один по левому, а другой по правому краю. Чем и хороши рисунки, что у них есть параметр *align*, который выравнивает изображение по тому краю, которому надо. Одновременно задавая у двух рисунков отбивку по левому и правому краю, получим нужный результат (Пример 6.1).

Пример 6.1. Выравнивание двух рисунков по сторонам

```


```


Однако, этот способ не лишен недостатков. Во-первых, чтобы добавить после рисунков текст, придется вставлять переносы строк. И во-вторых, выравнивание по вертикали всегда происходит только по верхнему краю. Более широкие возможности предоставляют в этом смысле таблицы, позволяя устанавливать изображения относительно друг друга в любом порядке.

Пример 6.2. Использование таблицы для выравнивания

```
<table width="100%" border="0" cellspacing="0" cellpadding="4">
<tr>
<td valign="bottom">
</td>
<td align="right" valign="bottom"></td>
</tr>
</table>
```



Рис. 6.1. Выравнивание элементов с помощью таблицы

Положение по вертикали изменяется параметром *valign* тега TD. Благодаря ему, объекты можно выравнивать по верхнему, нижнему краю либо по центральной оси. Атрибут *align="left"* писать необязательно, он задается по умолчанию. Рамка таблицы, естественно, делается невидимой толщиной ноль пикселей.

Требуется выровнять один объект по центру веб-страницы и одновременно другой по левому (или правому) краю. Такое размещение часто используется при публикации формул с нумерацией (рис. 6.2).

$$\left. \begin{aligned} \frac{dT_1}{ds} + \mu(T_1 - T_2) + k_1 N_1 + q_1 &= 0; \\ \frac{dN_1}{ds} + \mu N_1 - k_1 T_1 - k_2 T_2 + q_3 &= 0; \\ \frac{dM_1}{ds} + \mu(M_1 - M_2) &= N_1. \end{aligned} \right\} \quad (2.7)$$

Рис. 6.2. Выравнивание формулы и ее номера

Для такого размещения элементов, требуется создать таблицу с тремя ячейками. Крайние из них должны иметь одинаковые размеры, в средней ячейке выравнивание делается по центру, а в правой, понятное дело, по правому краю (пример. 6.3).

Пример 6.3. Выравнивание по центру и краю

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="10%">&nbsp;   </td>
<td align="center"></td>
<td align="right" width="10%">(2.7)</td>
</tr>
</table>
```

Точно также при необходимости помещается текст и в левую ячейку.

Выравнивание по вертикали

И на этот раз таблицы приходят нам на помощь, создавая невидимую модульную сетку, в ячейках которой можно размещать любые элементы. Располагать, причем, можно по разному, например, как показано ниже. В данном случае, заголовки располагаются на одном уровне, а текст на другом, как показано на рис. 6.3.

Как сидеть	<i>При работе на вычислительной технике необходимо сесть так, чтобы руки с предплечьями образовывали прямой угол, глаза поставить на расстояние 30-40 см от рабочей поверхности монитора.</i>
Как набирать	<i>Набирать на клавиатуре следует подушечками пальцев короткими отрывистыми ударами. Предельно допустимая длина ногтей для женщин составляет 12-15 мм, для мужчин 3-5 мм.</i>

Рис 6.3. Выравнивание элементов по вертикали

Код для данного примера приведен ниже.

Пример 6.4. Выравнивание по вертикали

```
<table width="100%" cellspacing="0" cellpadding="0">
<tr><td colspan="2"><b>Как сидеть</b></td></tr>
<tr bgcolor="#000000">
<td colspan="2"></td>
</tr>
<tr>
<td width="20%">&nbsp;</td>
<td><i>При работе на вычислительной технике необходимо сесть
так, чтобы руки с предплечьями образовывали прямой угол,
глаза поставить на расстояние 30-40 см от рабочей поверхности
монитора.<br></i></td>
</tr>
<tr><td colspan="2"><b>Как набирать</b></td></tr>
<tr bgcolor="#000000">
<td colspan="2"></td>
</tr>
<tr>
<td width="20%">&nbsp;</td>
<td><i>Набирать на клавиатуре следует подушечками пальцев
короткими отрывистыми ударами. Предельно допустимая длина
ногтей для женщин составляет 12-15 мм, для мужчин 3-5
мм.</i></td>
</tr>
</table>
```

Точно также можно размещать не только текст, но и рисунки и другие объекты.

7. ВЕРТИКАЛЬНЫЕ ЛИНИИ

Вертикальные линии или как их еще называют на типографский манер, вертикальные линейки, прекрасный декоративный элемент, который находит применение в множестве разных случаев. Помещенная возле текста вертикальная линия привлекает к нему внимание читателя, позволяя тем самым выделять нужные абзацы или блоки текста. Вертикальные линии могут также служить для разделения колонок текста, чтобы взгляд читателя не перескакивал с одной колонки на другую и четко отделить одну колонку от другой.

Для создания вертикальной линии фиксированной высоты можно использовать изображение необходимого цвета размером 1 на 1 пиксел. Почему именно 1 на 1? Да это очень удобно. Размер файла ничтожно мал, зато его можно масштабировать как угодно (Пример 7.1).

Пример 7.1. Линия сделанная с помощью рисунка 1 на 1 пиксел

```

или для указания относительной высоты можно указать ее в процентах


```

В примере, файл с именем *red1x1.gif* представляет собой рисунок, состоящий из одного пиксела красного цвета.

Для изменения цвета такой линии придется делать новый рисунок в каком-нибудь графическом редакторе. Кроме того, высота линии фиксирована и ее сложно привязать к тексту, чтобы она изменялась вместе с ним. В этом случае, на помощь нам придет таблица. Создаем таблицу, одна из ячеек которой имеет ширину 1-2 пиксела. Параметры *cellpadding* и *border* должны быть равны нулю, а *cellspacing* определяет расстояние от текста до линии. А затем в этой ячейке указываем фон нужного цвета: `<td width="1" bgcolor="#ff0000">`

Чтобы такая линия была видна в браузере Netscape, в ячейку нужно что-нибудь поместить. Неразделяемый пробел ` `; для этой цели не очень годится, он может сделать линию слишком толстой, что нам не нужно. Самое идеальное – прозрачный рисунок размером 1 на 1 пиксел (в примере 7.2 он именуется *spacer.gif*).

Пример 7.2. Линия созданная с помощью таблицы

```
<table width="600" cellpadding="0" cellspacing="12" border="0"
align="center">
<tr>
<td width="4" bgcolor="#990000"></td>
<td>Каждый из рефлексирующих субъектов виртуального мира может
находиться в двух основных состояниях: стационарном и мобильном, при
котором они или сводят передвижение в трехмерном Евклидовом
пространстве с декартовой системой координат к абстрактному
минимуму, или ограничиваются формальными перемещениями массы в
некоторой части пространственного континуума.</td>
</tr>
</table>
```

Каждый из рефлексирующих субъектов виртуального мира может находиться в двух основных состояниях: стационарном и мобильном, при котором они или сводят передвижение в трехмерном Евклидовом пространстве с декартовой системой координат к абстрактному минимуму, или ограничиваются формальными перемещениями массы в некоторой части пространственного континуума.

Рис. 7.1. Результат выполнения примера 7.2.

Этим же самым способом можно воспользоваться для создания декоративных линий, например, как на рисунке 7.2 или 7.3.

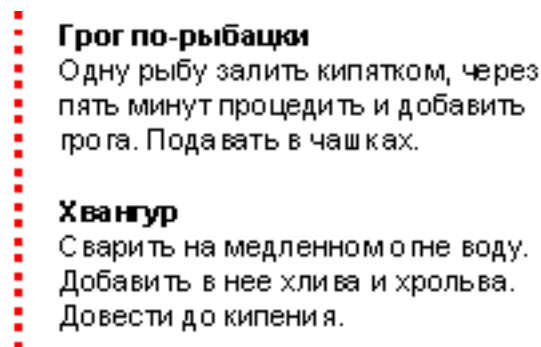


Рис. 7.2.

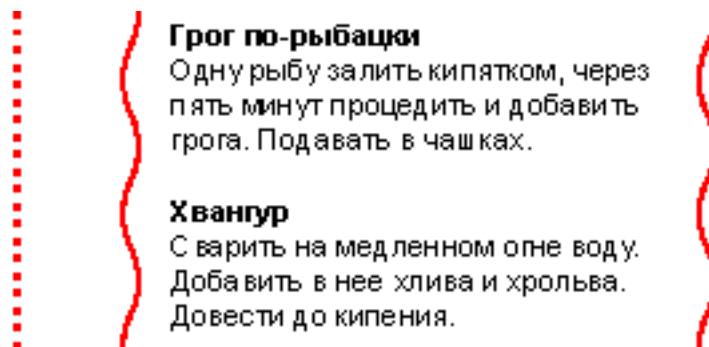


Рис. 7.3.

Для этого в ячейку таблицы помещаем фоновый рисунок с помощью параметра *background* тега TD.

Пример 7.3. Декоративные вертикальные линии с помощью таблицы

```
<table width="600" cellpadding="0" cellspacing="12" border="0"
align="center">
<tr>
<td width="4" background="line.gif"></td>
<td>Каждый рефлексирующий субъект обладает набором знаний и имеет
позитивные возможности оказания влияния на информационное поле для
взаимодействия с субъектами виртуального мира, но в то же время
наделен плюралистическим мировоззрением, позволяющим ему при
необходимости достижения или сохранения собственных материальных
благ пренебрегать законами логики и здравого смысла.</td>
</tr>
</table>
```

Файл *line.gif* представляет собой изображение линии, как, например, показанное на рис. 2.5.

8. ГОРИЗОНТАЛЬНЫЕ ЛИНИИ

Горизонтальные линии, или, на типографский манер, горизонтальные линейки хорошо использовать для отделения одного блока текста от другого. Небольшой по размеру текст, сверху и снизу которого располагаются горизонтальные линии привлекает больше, чем обычно, внимание читателя.

Если нужна простая, без затей линия, проще воспользоваться тегом HR.

Пример создания горизонтальных линий с помощью тега HR

Такая линия, однако, может иметь отличия в разных браузерах.

Пример 8.1. Создание обычных линий с помощью тега HR

```
<hr>  
Пример текста с обычными линиями  
<hr>
```

По умолчанию линия отрисовывается трехмерная, что не всегда вяжется с дизайном сайта. Поэтому, если требуется сделать сплошную линию, следует добавить параметр *noshade* тега HR.

Пример 8.2. Создание сплошной линий

```
<hr noshade size="2" width="300">  
Пример создания сплошных линий  
<hr noshade size="400">
```

Параметр *size* определяет толщину линии, а *width* ее длину в пикселах или процентах. Такая линия всегда размещается по центру экрана, если ее выравнивание специально не задается.

В качестве горизонтальных линий также можно использовать любое подходящее изображение, надо лишь растянуть его до нужной ширины. Ниже, в качестве рисунка, использовалась нарисованная буква Н.



Рис. 8.1. Горизонтальная линия из рисунка

Пример 8.3. Создание горизонтальных линий из рисунка

```
<img src=line.gif width=100% height=10><br>  
Пример линии сделанной из рисунка 100% ширины  
  
или  
  
<img src=line.gif width=400 height=10><br>  
Пример линии сделанной из рисунка фиксированной ширины 400 пикселей.
```

При таком растягивании рисунка в нем однозначно появляются искажения. Следовательно, чем абстрактней рисунок, тем лучше; чтобы никто ничего не понял, что на нем изображено.

Что делать, если нужна не сплошная линия, а, например, пунктирная? Для этого следует воспользоваться таблицей. Создаем таблицу состоящую из одной ячейки и вставляем в нее фоновый рисунок в виде пунктирной линии. Хитрость состоит в том, чтобы в ячейке кроме фона был еще один рисунок, иначе в браузере Netscape мы просто ничего не увидим. В качестве такого рисунка традиционно используется прозрачный GIF размером 1 на 1 пиксел.

Пример 8.4. Создание горизонтальных линий с помощью таблиц

```
<table border="0" background="dot.gif" width="100%" height="2"
cellspacing="0" cellpadding="0">
<tr>
<td></td>
</tr>
</table>
```

Файл *dot.gif* заранее готовится в графическом редакторе и может представлять, например, вот такой вот рисунок 

Еще один прием создания линий основан на использовании стилей. Вместо таблицы, описанной в предыдущем примере, используется тег DIV:

```
<div class="dot">&nbsp;</div>
```

Сам класс *dot* описан в таблице стилей следующим образом:

```
.dot { width: 100%; background: URL(dot.gif) repeat-x; }
```

где *URL* определяет имя рисунка, а *width* – ширину линии.

Можно в описание класса заложить еще и вертикальные отступы, что тоже очень удобно: *margin-top: 10px; margin-bottom: 10px;*

Пример 8.5. Создание горизонтальных линий с помощью стилей

```
<html>
<head>
<style type="text/css">
.dot { width: 100%; background: URL(dot.gif) repeat-x; }
</style>
</head>
<body>
<div class="dot">&nbsp;</div>
</body>
</html>
```

Для создания линии с помощью стилей можно также воспользоваться параметром *border-bottom*. Возможные виды линий показаны на рис. 3.6.

Пример 8.6. Создание горизонтальных линий с помощью стилей

```
<html>
<head>
<style type="text/css">
.dot { width: 100%; border-bottom: 1px dashed red }
</style>
</head>
<body>
<div class="dot">&nbsp;</div>
</body>
</html>
```

В примере создается пунктирная линия красного цвета толщиной один пиксел.

9. СОЗДАНИЕ КОЛОНОК

Две и более колонки позволяют разбить длинный текст или другое содержимое на блоки. В бумажных изданиях разбиение текста на колонки продиктовано сокращением ширины текста, поскольку для человека читать строку длиной более 60 символов утомительно. При чтении на бумаге читатель может охватить сразу весь текст и легко перейти с конца первой колонки на начало второй. На веб-странице такой подход не применяется, ведь придется мотать скроллбар вверх-вниз, что несколько неудобно. Вот поэтому создание колонок на сайте нужно для размещения разных независимых блоков. В левой, например, рубрикатор, в правой — новости.

Колонки легко делаются с помощью таблицы с двумя ячейками. Надо только задать параметр таблицы *cellpadding* или *cellspacing* отличным от нуля, чтобы сохранить расстояние между колонками.

Пример 9.1. Создание двух колонок с помощью таблицы

```
<table border="0" cellpadding="10" cellspacing="0">
<tr>
  <td valign="top">1 колонка</td>
  <td valign="top">2 колонка</td>
</tr>
</table>
```

По умолчанию содержимое ячейки выравнивается по центру вертикали, поэтому чтобы в колонках все было выровнено по верхнему краю, следует использовать параметр ячейки *valign="top"*.

Какой параметр выбрать – *cellpadding* или *cellspacing* зависит от вкуса и желания. В данном случае абсолютно никакой разницы нет.

10. КОЛОНКИ С РАЗДЕЛИТЕЛЕМ

Две и более колонки позволяют разбить длинный текст или другое содержимое на блоки. Такие блоки разделяют между собой вертикальной линией, чтобы взгляд читателя не перескакивал с одной колонки на другую.

Колонки с разделителем создаются с помощью таблицы с тремя ячейками. Среднюю ячейку делаем шириной 1-2 пиксела и заливаем каким-нибудь цветом с помощью параметра *bgcolor* – вот и получится линия. А параметр *cellspacing* определяет расстояние между линией и текстом. Использовать *cellpadding* в данном случае нельзя, иначе линия окажется толще, чем необходимо.

Пример 10.1. Создание двух колонок с линией между ними

```
<table border="0" cellpadding="0" cellspacing="10">
<tr>
  <td valign="top">1 колонка</td>
  <td width="1" bgcolor="#000000"></td>
  <td valign="top">2 колонка</td>
</tr>
</table>
```

Чтобы разделительная линия была видна в браузере Netscape, в ячейку нужно что-нибудь поместить. Неразделяемый пробел ` ` для этой цели не годится, он может сделать линию более толстой, что нам не нужно. Самое идеальное - прозрачный рисунок размером 1 на 1 пиксел (в примере он называется *spacer.gif*).

11. ПОДРИСУНОЧНАЯ ПОДПИСЬ

Подрисуночная надпись – текст, являющийся комментарием к рисунку и его описывающий. Такая подпись важна, поскольку привлекает внимание читателя к рисунку и дает больше информации об изображении. У тега `IMG` существует, конечно, параметр `alt`, который задает текст подсказки, но чтобы ее получить приходится наводить курсор мыши на каждый рисунок, что довольно неудобно. Более наглядный способ и, соответственно, более предпочтительный заключается в размещении подрисуночной надписи возле самого изображения. Подпись хоть и называется подрисуночной, но может располагаться и сбоку от рисунка, если это продиктовано соображениями верстки и дизайна.

Для размещения на веб-странице и рисунка и подписи к нему удобней всего воспользоваться таблицей. В этом случае можно задать цвет каемки вокруг рисунка, ее цвет, ширину, а также местоположение текста (рис. 11.1).

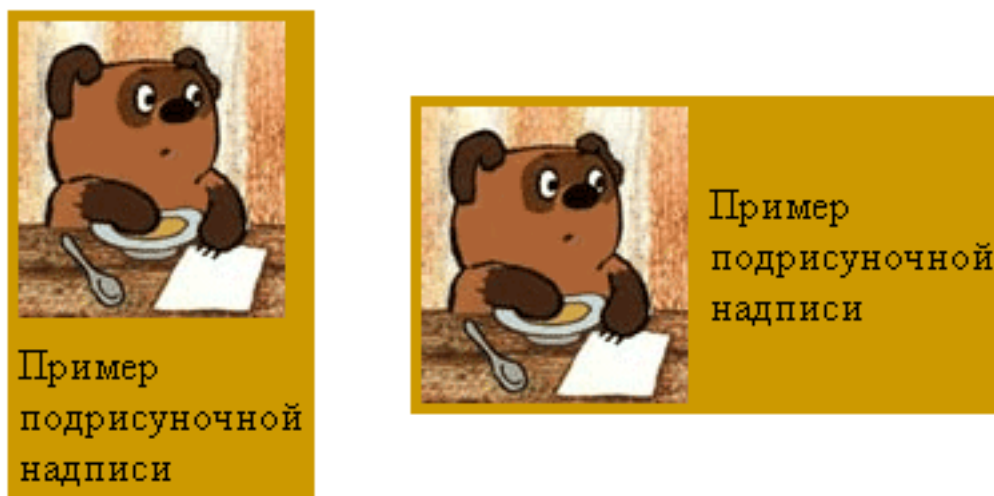


Рис. 11.1. Варианты размещения подрисуночной надписи

Пример 11.1. Создание подрисуночной надписи с помощью таблицы

```
<table width="100" border="0" cellspacing="0" cellpadding="4"
bgcolor="#CC9900" align="center">
<tr>
  <td></td>
</tr>
<tr>
  <td>Пример подрисуночной подписи</td>
</tr>
</table>
```

Ширина таблицы определяется исходя из размеров рисунка и желаний автора. В данном примере ширина таблицы совпадает с шириной рисунка; ширина цветной рамки вокруг изображения задается с помощью параметра *cellpadding* или *cellspacing*. В данном случае абсолютно все равно, что предпочесть, результат будет один. Цвет таблицы, указанный параметром *bgcolor*, определяет цвет подложки вокруг изображения и подписи к нему.

Выше представлен простой вариант, когда рисунок с подписью размещен по центру страницы. Если же необходимо поместить нашу конструкцию в текст чтобы он ее обтекал, в этом случае можно воспользоваться свойством таблицы *align=left | right*. Это свойство выравнивает таблицу по левому или правому краю страницы. Все бы ничего, да вот отступы от края таблицы до текста, как в теге `IMG`, не предусмотрены. Получается очень некрасиво, когда текст словно наезжает на наш рисунок. Страдают принципы дизайна и простой здравый смысл подсказывает, что читать такой текст ой как нелегко. Нам надо получить отступы вокруг таблицы, и для этого следует воспользоваться вложенными таблицами. Создаем таблицу, выровненную по левому или правому краю, той же ширины, что и рисунок, а параметр *cellpadding* будет определять расстояние от текста до таблицы.

Пример 11.2. Создание отступов вложенными таблицами

```
<table width="100" border="0" cellspacing="0" cellpadding="6"
align="left">
<tr>
  <td>
    <table width="100" border="0" cellspacing="0" cellpadding="4"
bgcolor="#CC9900">
      <tr>
        <td></td>
      </tr>
      <tr>
        <td>Пример подрисуночной надписи</td>
      </tr>
    </table>
  </td>
</tr>
</table>
```

12. СОЗДАНИЕ БУКВИЦЫ

Буквица является художественным приемом оформления текста и представляет собой увеличенную первую букву, базовая линия которой ниже на одну или несколько строк базовой строки основного текста. Буквица привлекает внимание читателя к началу текста, особенно если страница лишена других ярких деталей. Обычно кроме самого символа используются изображения растений, животных и других объектов. Это конечно не обязательно, но может придать определенный настрой содержанию. Если хочется именно так сделать на сайте, лучше всего для этого подойдет рисунок выровненный по левому краю.

Пример 12.1. Создание буквицы с помощью рисунка

```
еобходимо, манипулируя полученными предметами
материального мира и фрагментами информационного поля, эмпирическим
путем достигнуть логического завершения конкурса.
```

Отступ от буквицы до текста задается параметрами *vspace* – по вертикали и *hspace* – по горизонтали.

Достоинства использования рисунков в качестве буквицы следующие: применение любого шрифта и разных эффектов, простота метода, а также возможность использования в качестве формата анимированный GIF, что дает дополнительные возможности оформления. Недостатки: если буквица на сайте применяется довольно часто, придется подготовить множество рисунков разных букв; усложняется возможность редактирования текста, так как придется вместо простого изменения буквы вставлять другой рисунок. Ниже описано, как делать буквицу текстом.

Для того, чтобы текст огибал первую букву, как показано на рис. 12.1, ее необходимо поместить в ячейку таблицы, выровненную по левому краю.



Рис. 12. 1. Буквица в тексте

Пример 12.2. Буквица в виде текста

```
<table align="left" border="0" cellspacing="0" cellpadding="0">
<tr><td><font size="7">"Д</font></td></tr>
</table>имон, у тебя есть что-нибудь похожее на трансплюкатор?". "Что
значит похожее?!" - восклицает в ответ Дима - "У меня есть самый
настоящий трансплюкатор!". С этими словами лезет на балкон и достает
оттуда такую штуку, при виде которой сразу понимаешь - Он.
```

При желании можно изменить цвет фона и рамки параметрами *bgcolor* и *bordercolor*. Отступы от буквицы до текста управляются значением *cellpadding*

или *cellspacing*. Для изменения размера и шрифта символа также используются стили. В этом случае код HTML будет незначительно отличаться.

Пример 12.3. Использование стилей

```
<html>
<head>
<style type="text/css">
.letter { font-family: Arial; font-size: 24pt; color: navy }
</style>
</head>

<body>
<table align="left" border="0" cellspacing="0" cellpadding="0">
<tr><td><span class="letter">"Д</span></td></tr>
</table>имон, у тебя есть что-нибудь похожее на транплюкатор?". "Что
значит похожее?!" - восклицает в ответ Дима - "У меня есть самый
настоящий транплюкатор!". С этими словами лезет на балкон и достает
оттуда такую штуку, при виде которой сразу понимаешь - Он.
</body>
</html>
```

Ничто не мешает расположить буквицу слева, четко отделив ее от основного текста. Таким образом можно добиться простого выразительного эффекта (рис. 12.2).



Рис. 12.2. Буквица на поле

Для размещения буквицы этим способом необходимо создать таблицу с двумя ячейками, в левой расположить первый символ, а в правой остальной текст.

Пример 12.4. Буквица на поле

```
<table width="100%" border="0">
<tr>
<td valign="top">
<font size="7">В</font>
или
<span class="letter">В</span>
или

</td>
<td valign="top">ыдалась у меня свободная минутка, открыл я любимый
LINES , сижу, строю пятерки и тихо наслаждаюсь. Заходит девочка и
начинает меня строить: дескать, LINES - это игра детсадовская, и
т.п. "А ты во что играешь?", - спрашиваю очаровательную девочку.
"Конечно, в Quake", - говорит она, - "там такое мясо"!
</td>
</tr>
</table>
```

Обратите внимание, содержимое обеих ячеек выравнивается по верхнему краю параметром *valign="top"*.

13. ОТСТУП ПЕРВОЙ СТРОКИ

Использование красной строки с отступом в 2-5 пробела повышает читаемость текста, позволяя легко отыскивать взглядом начало следующего абзаца. В HTML любое количество пробелов заменяется одним, в этом случае стоит использовать другой символ – ` `:

Пример 13.1. Создание отступа неразделимым пробелом ` `:

```
&nbsp; &nbsp; &nbsp; &nbsp; Сварить на медленном огне воду. Положить в нее  
небольшого червяка, лучше живого, но сойдет и так, мелко покрошенное  
яблоко, крылышко мотылька и жабу. Соль и сахар по вкусу. По  
готовности перелить в бутылки. Хранить в прохладном месте.
```

Следующий способ также имеет право на существование. Вместо символов пробела надо поставить невидимый рисунок нужной ширины.

Пример 13.2. Создание отступа прозрачным рисунком

```
Сварить на медленном огне  
воду. Добавить в нее хлива и хрольва. Довести до кипения.
```

В качестве рисунка можно использовать прозрачный рисунок в формате GIF размером 1 на 1 пиксел.

И, наконец, всегда можно воспользоваться стилями. Параметр *text-indent* задает отступ первой строки текста. Отступ можно указывать в пунктах (pt), пикселах (px), дюймах (in), миллиметрах (mm) и др.

Пример 13.3. Создание отступа с помощью стилей

```
<html>  
<head>  
<style type="text/css">  
p { text-indent: 20 px }  
</style>  
</head>  
<body>  
<p>Смешать 2 части соляной кислоты и 1 часть азотной со льдом. Слить  
охлажденную смесь в фужер. Пить залпом.</p>  
</body>  
</html>
```

Конечно, использование стилей более универсально, вдобавок менее обременительно. Однако применение символов ` ` является более простым, надежным и независимым от браузера способом. При использовании же в качестве отступа невидимого рисунка есть опасность, что пользователь отключил загрузку изображений, тогда вместо отступа будет показываться некрасивая полоса. В конечном итоге, какой метод предпочесть – решать вам.

14. ОТСТУПЫ НА ВЕБ-СТРАНИЦЕ

Горизонтальные и вертикальные отступы от края браузера до содержимого веб-странице встроены в браузер по умолчанию. Тем не менее можно изменять значение этих параметров, делая отступы по желанию больше или меньше. Хитрость заключается в том, что Internet Explorer и Netscape имеют разные параметры для указания значения отступов. У Internet Explorer в теге BODY следует указывать параметр *leftmargin* для горизонтального отступа и *topmargin* для вертикального, а в Netscape те же функции выполняют параметры *marginwidth* и *marginheight*. Объединяя все параметры воедино, получим универсальный код, который будет работать во всех браузерах одинаково.

Пример 14.1. Изменение величины отступов

```
<body bgcolor="#FFFFFF" text="#000000" leftmargin="20"  
topmargin="20" marginwidth="20" marginheight="20">
```

Частенько отступы вообще убирают, делая их равными нулю. Тогда рабочее поле становится больше размером, а используемые рисунки и таблицы, выравненные по краю, располагаются «под обрез». Ниже приведен пример создания таблицы 100% ширины без пустого пространства между краем браузера и таблицей.

Пример 14.2. Отсутствие отступов на веб-странице

```
<html>  
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0"  
marginwidth="0" marginheight="0">  
  
<table width="100%" border="0" cellspacing="0" cellpadding="0"  
height="100">  
<tr>  
<td bgcolor="#CC6600">&nbsp;  </td>  
</tr>  
</table>  
</body>  
</html>
```