

Многоколоночная верстка

Руководство web мастера

Содержание

Вступление	3
Для кого это руководство.....	3
Создаем простую страницу.....	4
Размещаем блоки в две колонки.....	6
Меняем колонки местами.....	11
Верстка в две колонки с фиксированной шириной.....	12
Добавляем третью колонку.....	14
Создаем разделители между колонками.....	15
Заключение.....	19
Полезные ссылки.....	20

Об авторе. Перед вами руководство, написанное автором блога **www.simplecoding.org** Стаценко Владимиром. Пожалуйста, все вопросы, замечания и пожелания отправляйте на eMail **vvv_33@gala.net**.

Лицензия. Копирование, распространение, демонстрация, модификация и другое использование данного руководства разрешается только в некоммерческих целях и при условии размещения ссылки на сайт автора. Коммерческое использование возможно только с письменного разрешения автора.

Вступление

Вы интересуетесь созданием web сайтов? Хотите сделать свой собственный сайт (блог)? Не просто страничку, размещенную на каком-нибудь бесплатном сервисе, и похожую на тысячи других, а современный сайт с уникальным дизайном, сотнями страниц, удобной панелью управления?... Если да, то вам предстоит довольно большой фронт работ :-). И, надеюсь, это руководство немного упростит вашу задачу.

Мы рассмотрим способы размещения содержимого web страниц в несколько колонок. А также разберем, как можно управлять их размещением, не изменяя разметку страницы.

Приведенные в руководстве примеры и рекомендации отвечают следующим требованиям:

- страницы одинаково отображаются во всех современных браузерах (Internet Explorer, Firefox, Opera);
- соответствуют web стандартам (XHTML 1.0, CSS 1.0);
- позволяют изменять дизайн страницы, не изменяя html разметку.

Для кого это руководство

Все приведенные примеры сопровождаются довольно подробным описанием. Тем не менее, от вас потребуются базовые знания HTML и CSS. Например, если вы можете сказать в чем разница между `<body>`, `<p>`, `#title` и `.title`, то спокойно разберетесь во всех примерах.

Основные способы верстки web страниц

На сегодняшний день широкое распространение получили два способа верстки:

1. верстка с помощью html таблиц;
2. "блочная" верстка.

Первый способ предполагает создание большой html таблицы, в ячейках которой размещаются элементы страницы (текст, изображения, меню и т.п.).

При "блочной" верстке фрагменты страницы обычно размещают внутри тегов `<div>`, а их взаимное расположение задается с помощью правил в каскадной таблице стилей (CSS).

О том, какой из этих способов лучше, web мастера спорят до сих пор. Я не собираюсь отстаивать чью-либо позицию, просто хочу обратить ваше внимание на несколько, на мой взгляд, ключевых преимуществ "блочной" верстки.

1. Возможность изменения внешнего вида страницы с помощью таблицы стилей (html разметка остается прежней).

2. Читать HTML разметку при табличной верстке гораздо тяжелее. Главным образом из-за большого количества тегов <td>. А это усложняет создание динамических страниц. Поверьте, вставка PHP (или любого другого) кода в такую страницу далеко не самое приятное занятие.
3. Блочная верстка позволяет создавать несколько версий дизайна страницы для различных устройств. Например, при просмотре с помощью обычного браузера будут отображаться фоновые изображения, а при распечатке на принтере – нет.

Кроме того, вы можете использовать оба способа верстки одновременно. Так что, выбор за вами.

Создаем простую страницу

Пора переходить к делу. Создадим простую страницу, состоящую из четырех блоков:

- заголовка (header);
- основной части (main);
- меню (menu);
- и "хвостовика" (footer).

Разместить эти блоки нам нужно так, как показано на рисунке 1.



Рис. 1.

Как видите, это обычное размещение, которое используется большинством блогов и небольших сайтов.

Теперь перейдем к html разметке:

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">
02 <html xmlns="http://www.w3.org/1999/xhtml" lang="ru">
03 <head>
04 <title>Страница с двумя колонками</title>
05 <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
06 <link rel="stylesheet" type="text/css" href="twocols.css" />
07 </head>
08
09 <body>
10 <div id="header">Заголовок </div>
```

```
11 <div id="main">Основной блок</div>
12 <div id="sidebar">Меню</div>
13 <div id="footer">Хвостовик </div>
14 </body>
15 </html>
```

Листинг 1.

Разберем по порядку, что мы здесь написали. В начале файла (строки 1,2) мы указали тип документа (разметка нашей страницы соответствует стандарту XHTML1.0 Strict).

Начиная со строки 3, идет разметка страницы. Тут нужно обратить внимание на следующее.

1. В строке 6 мы подключили файл с таблицей стилей (twocols.css).
2. Каждый блок нашей страницы заключен в теги <div> ... </div>. Причем, в атрибуте id этого тега мы указываем имена блоков. Это очень важный момент. Используя эти имена, мы будем управлять размещением блоков на странице. Кстати, чтобы листинг легче читался, я убрал из него содержимое блоков.
3. Блок main размещен выше menu. Для верстки страницы очередность размещения этих блоков роли не играет, но поисковые системы любят, когда содержимое страницы находится ближе к ее началу.

Создадим таблицу стилей для нашей страницы.

```
01 @charset "UTF-8";
02
03 #header {
04     border: solid 1px #ff0000;
05     margin: 10px 10px 10px 10px;
06     padding: 10px 10px 10px 10px;
07 }
08
09 #main {
10     border: solid 1px #ff0000;
11     margin: 10px 10px 10px 10px;
12     padding: 10px 10px 10px 10px;
13 }
14
15 #sidebar {
16     border: solid 1px #ff0000;
17     margin: 10px 10px 10px 10px;
18     padding: 10px 10px 10px 10px;
19 }
20
21 #footer {
22     border: solid 1px #ff0000;
23     margin: 10px 10px 10px 10px;
24     padding: 10px 10px 10px 10px;
25 }
```

Листинг 2.



Обратите внимание на кодировку, объявленную в обоих файлах (UTF-8). Вы можете использовать любую другую, но не забудьте проверить, что она совпадает с кодировкой файла.

Тут мы добавили рамки (`border`) вокруг каждого блока и установили отступы между блоками (`margin`) и внутри них (`padding`). Обратите внимание на то, как заданы стили. Сначала указывается селектор блока (например, `#menu`). А за ним, в фигурных скобках, правила. Каждое правило состоит из свойства и значения, разделенных двоеточием.

Границу мы нарисовали красной (`#ff0000`) сплошной (`solid`) линией толщиной `1px`. А для всех отступов задали значение `10px`. Кстати, отступы задаются в следующем порядке: верхний, правый, нижний, левый.



Правило `border: solid 1px #ff0000`; является одним из основных средств отладки CSS кода.

Теперь посмотрим, как выглядит страница на данном этапе.

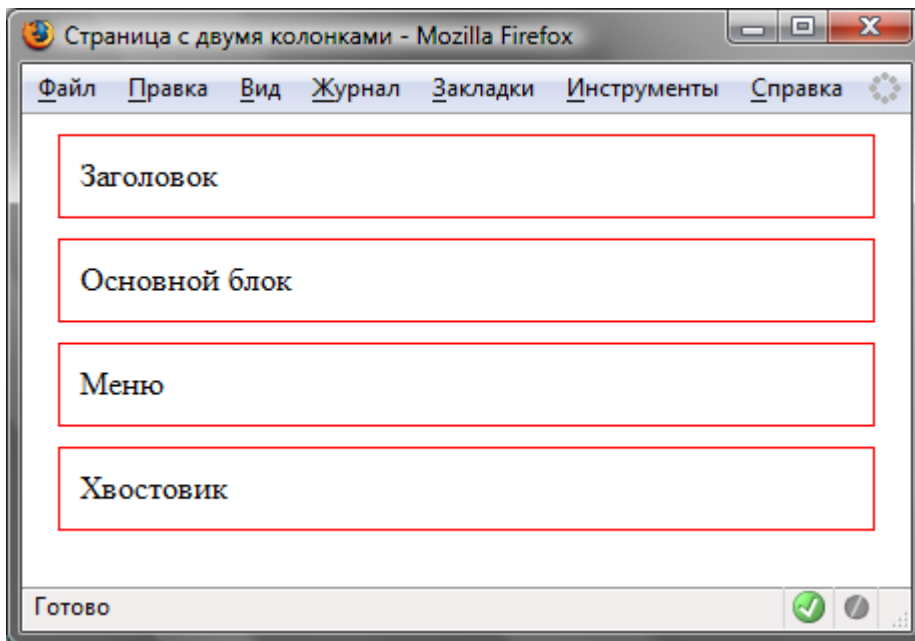


Рис.2.

В общем, так как и следовало ожидать. Блоки расположены друг за другом в порядке размещения в `html` файле.

Размещаем блоки в две колонки

Расположим меню (`sidebar`) справа от основного блока (`main`), т.е. разместим их в две колонки.

Как я и говорил, нам потребуется изменить только таблицу стилей. HTML файл останется без изменений.

Многоколоночная верстка. Руководство web мастера

Прежде всего, определимся с шириной колонок. Допустим, блок `main` должен занимать 70% ширины экрана, а `sidebar` – все остальное.

Для размещения в две колонки мы используем свойство `float` (подробнее о нем чуть ниже), а для установки ширины колонок – свойство `width`.

Добавим эти правила в таблицу стилей:

```
01 #main {
02     border: solid 1px #ff0000;
03     margin: 10px 10px 10px 10px;
04     padding: 10px 10px 10px 10px;
05
06     float: left;
07     width: 70%;
08 }
09
10 #sidebar {
11     border: solid 1px #ff0000;
12     margin: 10px 10px 10px 70%;
13     padding: 10px 10px 10px 10px;
14 }
15 }
```

Листинг 3.

Обратите внимание, что мы увеличили отступ справа для блока `sidebar` (строка 12) так, чтобы его хватало для размещения блока `main`.

Посмотрим, что получилось

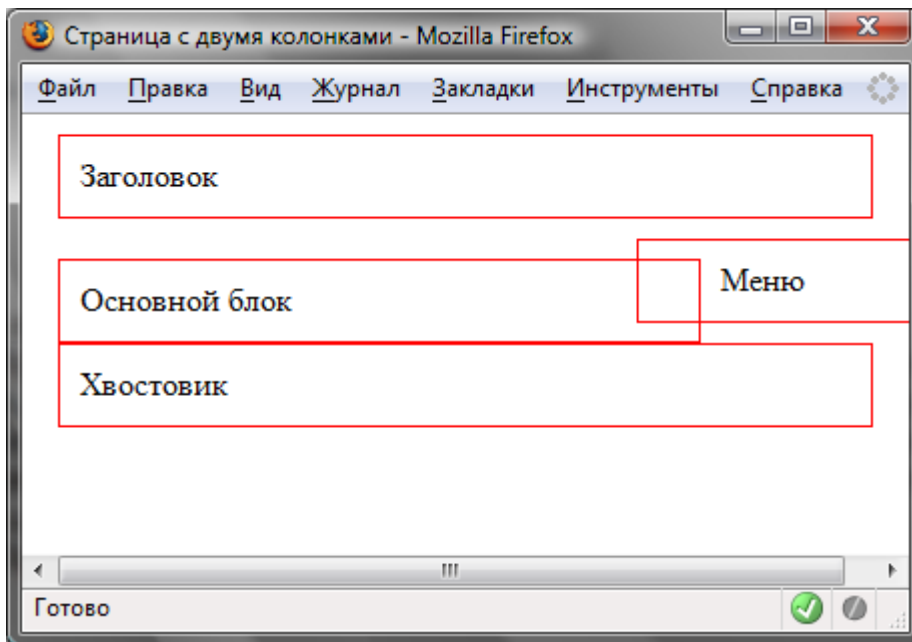


Рис. 3.

Как видите, колонок две, но блоки перекрываются, и появилась горизонтальная полоса прокрутки. Попробуем разобраться почему. Мы

Многоколоночная верстка. Руководство web мастера

установили ширину основного блока и меню 70% и 30%, т.е. в сумме получается 100% (полная ширина окна браузера). Но для каждого блока у нас были заданы отступы величиной 10 px (строки 5, 11, 17, 23 в листинге 2). Таким образом, получается, что ширина нашей страницы равна: 70% + 30% + отступы.

Чтобы исправить ситуацию, мы уменьшим ширину блока main на 10%, а отступ слева от sidebar оставим неизменным.

```
#main {  
    ...  
    float: left;  
    width: 60%;  
}
```

Листинг 4.

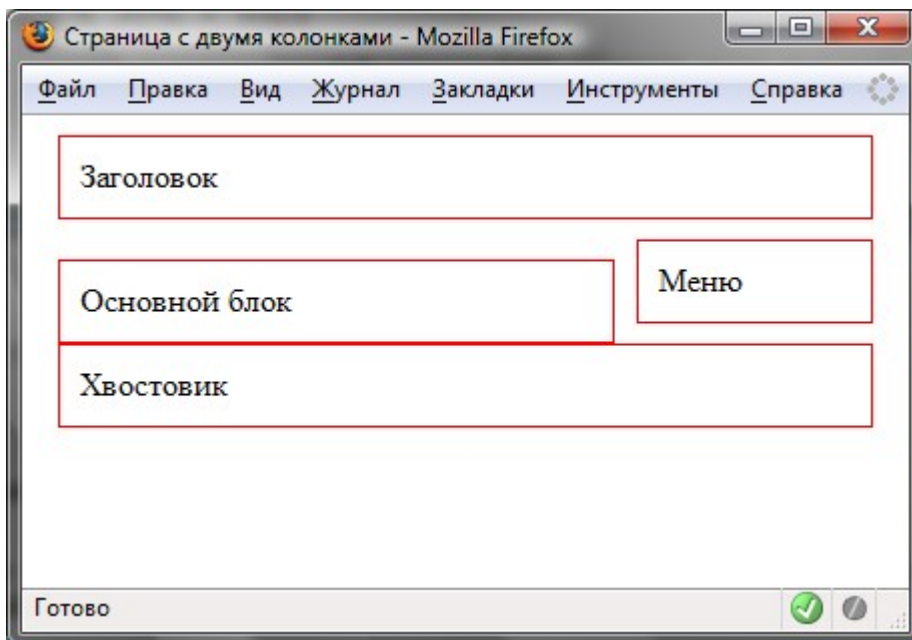


Рис.4.

Теперь страница выглядит гораздо лучше.

i Для того чтобы блоки не перекрывались, 10% от ширины страницы должно быть всегда больше, чем 10px. Т.е. если пользователь сильно уменьшит ширину окна браузера, блоки опять перекроются.

Кроме того, если вы откроете страницу в IE, то увидите, что основной блок смещен вправо на величину отступа (10px). Позже я покажу, как это исправить.

Разберем подробнее, как работает свойство `float`.

По-умолчанию, браузер отображает блоки в порядке их следования в html файле (этот порядок следования называется потоком документа).

Многоколоночная верстка. Руководство web мастера

Установка свойства float "вырывает" блок из потока. А значение свойства (left или right) указывает, что блок должен быть сдвинут влево или вправо.

Это означает, что меню будет размещено сразу после заголовка страницы, так, как будто бы основного блока нет вообще. Вторая колонка создается за счет установки ширины блоков и левого отступа меню (строки 6, 7 и 12 в листинге 3).

На рисунке 5 показано, как сдвигаются блоки после установки этих свойств.

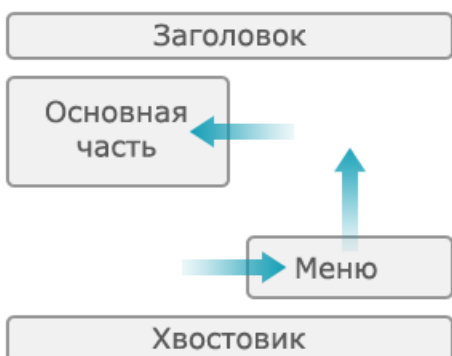


Рис.5.

Теперь посмотрим, что произойдет при увеличении длины блоков.

Для меню все просто. Увеличиваем его длину – footer смещается вниз.

Но посмотрите, что произойдет, если добавить несколько строк в основной блок.

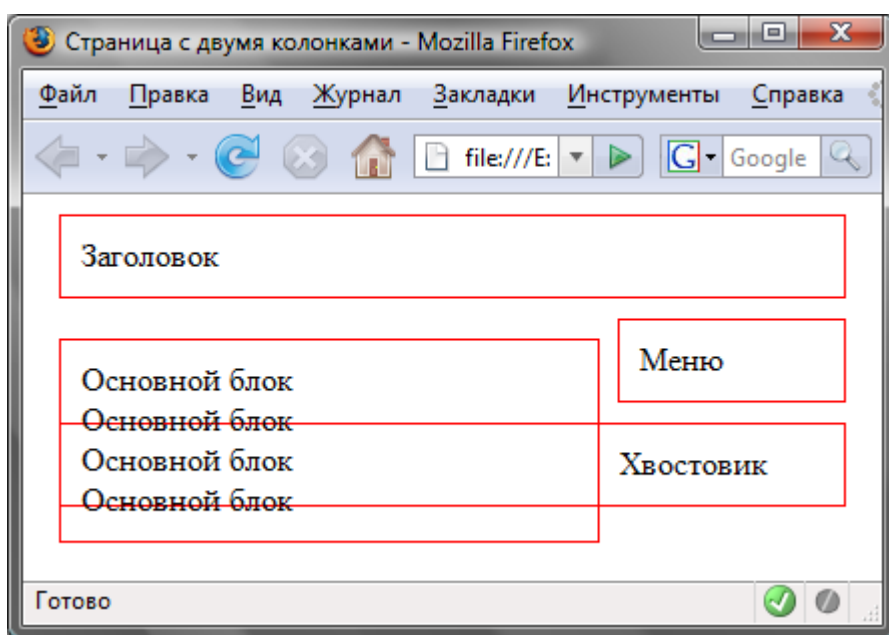


Рис.6.

Как и следовало ожидать, основной блок перекрывает хвостовик.

Исправить ситуацию можно установкой правила `clear: left`.

```
01 #footer {  
02     border: solid 1px #ff0000;  
03     margin: 10px 10px 10px 10px;  
04     padding: 10px 10px 10px 10px;  
05     clear: left;  
06 }
```

Листинг 5.

Оно дает указание браузеру разместить `footer` ниже всех блоков, для которых установлено правило `float: left`, т.е. ниже `main`. Результат его работы показан на рис.7.

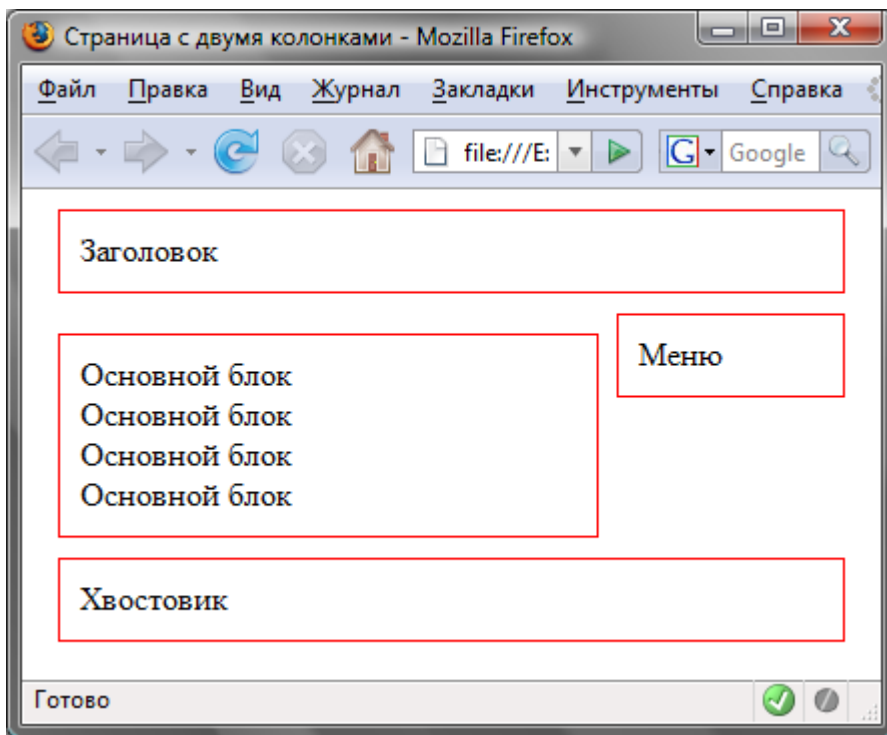


Рис.7.

Другие возможные значения свойства: `right` (блок будет размещен под всеми блоками, для которых установлено правило `float: right`) и `both` (под любыми блоками со свойством `float`).

Меняем колонки местами

Теперь попробуем немного изменить дизайн страницы. Поместим меню слева, а основной блок – справа. Для этого нам нужно изменить несколько правил в таблице стилей:

```
01 #main {
02     ...
03     float: right;
04     width: 60%;
05 }
06
07 #sidebar {
08     ...
09     margin: 10px 70% 10px 10px;
10     padding: 10px 10px 10px 10px;
11 }
12
13 #footer {
14     ...
15     clear: right;
16 }
```

Листинг 6.

Как видите, мы изменили, размещение блока `main` с помощью правила `float: right` (строка 3), т.е. сдвинули его вправо. А для меню мы поменяли значения правого и левого отступов местами (строка 9), и, таким образом, сдвинули его вправо.

Последнее изменение касается свойства `clear` в хвостовике (строка 15). Т.к. для блока `main` установлено правило `float: right`, значение `clear` тоже должно быть `right`.



Если для блока `footer` установить правило `clear: both`, то он всегда будет расположен под основным блоком, независимо от того, в какую сторону тот смещен.

На рис.8 показан результат наших усилий.

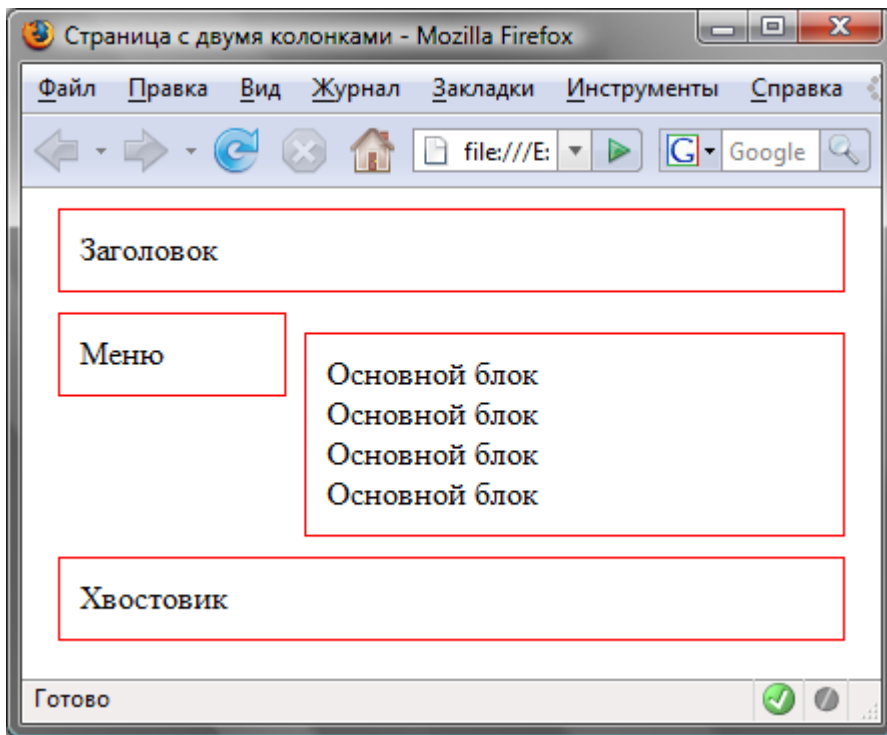



Рис.8.

	<p>Вы, наверное, обратили внимание, что граница основного блока находится ниже границы меню. Происходит это из-за того, что основной блок “вырван” из потока документа, и его верхний отступ прибавляется к верхней границе меню, а не заголовка.</p> <p>Избежать этого можно, установив для всех четырех блоков (<code>header</code>, <code>main</code>, <code>sidebar</code>, <code>footer</code>) отступы равными нулю (<code>margin:0; padding:0</code>). А для элементов, размещенных внутри этих блоков, задать стили с нужными отступами.</p>
---	--

Обратите внимание, что html разметка страницы осталась неизменной. Это, основное достоинство “блочной” верстки. Вы можете практически полностью изменить дизайн сайта только с помощью таблицы стилей. Если хотите посмотреть, насколько существенными могут быть изменения, зайдите на www.csszengarden.com.

Верстка в две колонки с фиксированной шириной

Страница, которую мы сделали, растягивается на всю ширину окна браузера. Если пользователь изменит его размер, то ширина обоих блоков изменится автоматически.

Это поведение можно изменить, если задать фиксированную ширину для блоков. Сделать это несложно. Достаточно указать ширину блоков в пикселях, а не процентах. Например, `width: 250px`.

Многоколоночная верстка. Руководство web мастера

Но, гораздо удобнее добавить внешний блок (`container`), ширина которого будет задана в пикселях, а стили внутренних блоков останутся прежними. Кроме того, такой подход позволяет легко отцентрировать страницу.

Посмотрите на разметку страницы.

```
01 <div id="container">
02   <div id="header">Заголовок</div>
03   <div id="main">Основной блок...</div>
04   <div id="sidebar">Меню</div>
05   <div id="footer">Хвостовик</div>
06 </div>
```

Листинг 7.

Как видите, мы разместили все наши блоки внутри нового блока (`container`). Теперь добавим стили для этого блока.

```
01 #container {
02   width: 400px;
03   border: solid 1px #ff0000;
04   margin: auto;
05 }
```

Листинг 8.

Здесь мы задали ширину (400px) и границу (исключительно для наглядности).

Правило `margin: auto`, которое мы установили в четвертой строке, выполняет центрирование блока. По сути, оно дает указание браузеру рассчитать левый и правый отступы от края окна до границы блока.

В результате мы получим что-то похожее на этот скриншот.

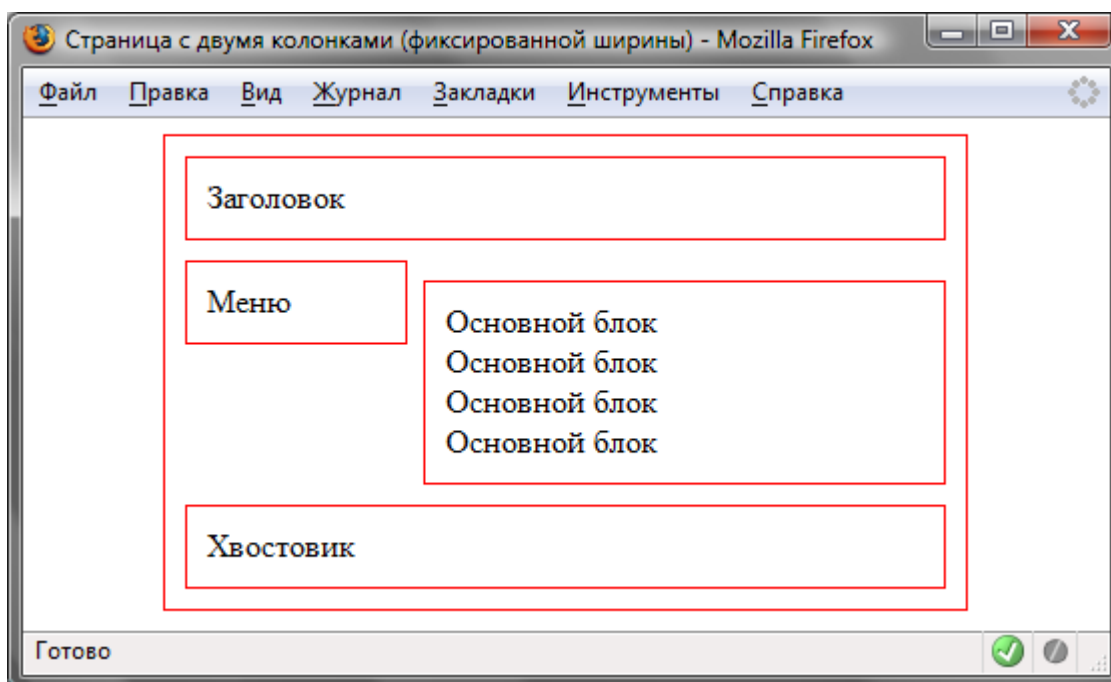


Рис. 9.

Добавляем третью колонку

Если вы прочли предыдущие разделы, то, наверное, уже догадались, как это сделать.

Прежде всего, создаем новый блок в html файле, назовем его `middlebar` (мы добавляем центральную колонку).

```
01 <div id="header">Заголовок</div>
02 <div id="main">Основной блок ... </div>
03 <div id="sidebar">Меню</div>
04 <div id="middlebar">Центральная колонка</div>
05 <div id="footer">Хвостовик</div>
```

Листинг 9.

Теперь внесем изменения в таблицу стилей. Для того чтобы поместить новую колонку по центру, сдвинем меню вправо, а блок `main` – влево (можно и наоборот, это не принципиально).

```
01 #main {
02     ...
03     float: right;
04     width: 30%;
05 }
06
07 #sidebar {
08     ...
09     float: left;
10     width: 20%;
11 }
12
13 #middlebar {
14     ...
15     margin: 10px 40% 10px 30%;
16 }
```

Листинг 10

Как видите, мы установили свойства `float` для блоков `main` и `sidebar`, одновременно уменьшив их ширину (строки 3, 4 и 9, 10). А для `middlebar` задали соответствующие отступы слева и справа.

Посмотрите на скриншот. Третья колонка расположена именно так, как и было задумано.

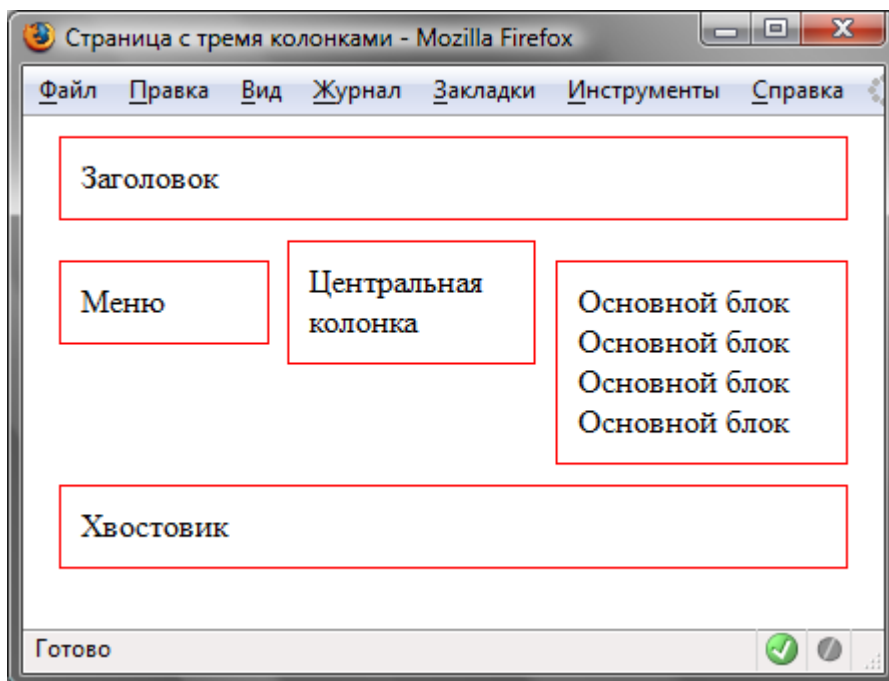


Рис. 10

Создаем разделители между колонками

Разделители используются в дизайне многих сайтов. Для их создания обычно используются фоновые изображения или свойство `border` блоков.

Но посмотрите на рисунок 10. Все наши колонки имеют разную высоту и, самое главное, она может изменяться. Например, центральная колонка может быть короче основного блока, а может быть и длиннее. А граница всегда будет той же высоты, что и блок.

Более того, нам нужны два разделителя. Между блоками `sidebar` и `middlebar`, и между `middlebar` и `main`. Причем оба разделителя должны быть одинаковой высоты.

Самый простой способ решить эту задачу – растянуть `middlebar` так, чтоб он всегда был длиннее любого из остальных блоков.

Для этого добавим внутрь `middlebar` еще один блок, и установим для него правило `clear: both`.

HTML разметка:

```
01 <div id="middlebar">
02 Центральная колонка
03 <div id="mfooter"></div>
04 </div>
```

Листинг 11

CSS разметка:

```
01 #mfooter {  
02     clear: both;  
03 }
```

Листинг 12

Результат показан на рисунке 11.

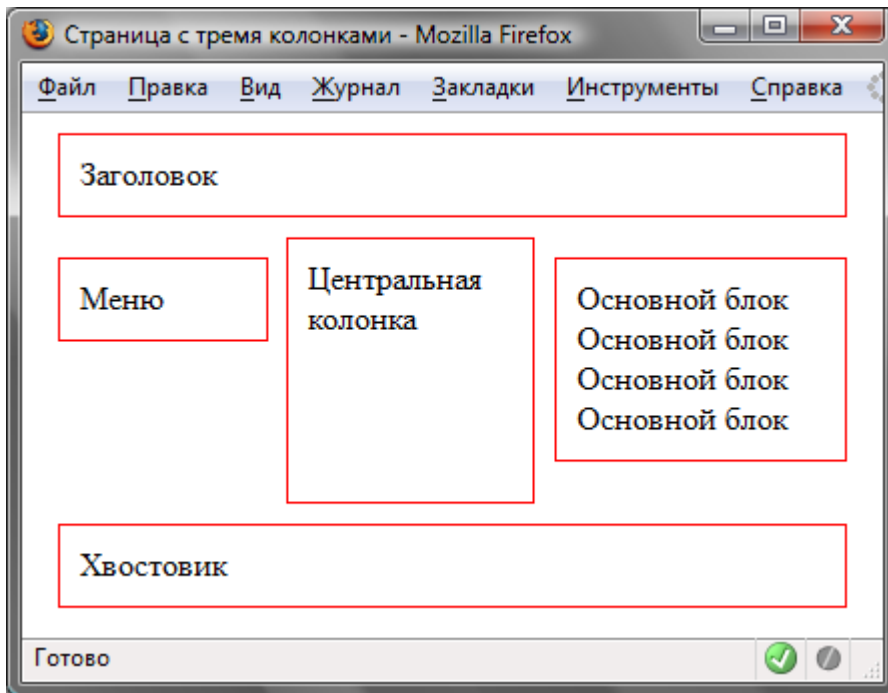


Рис.11

Разберемся, за счет чего растянулась колонка. Для блоков `sidebar` и `main` установлены свойства `float`. Это означает, что они “вырваны” из потока документа, а значит, блок `middlebar` должен начинаться сразу после заголовка.

Теперь рассмотрим блок `mfooter`. Для него установлено правило `clear: both`, следовательно, он должен быть расположен *ниже* всех блоков со свойством `clear`, которые размещены до него на странице. С другой стороны этот блок находится внутри `middlebar`. Таким образом, `middlebar` должен начинаться от заголовка и заканчиваться ниже блоков `sidebar` и `main`.

На рис.12 схематически показано размещение блоков.

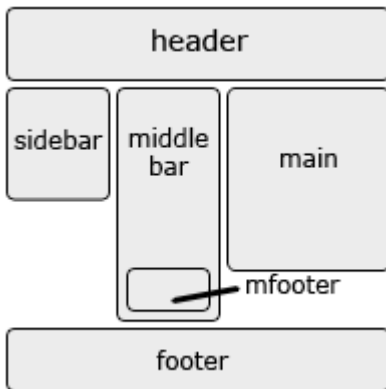


Рис.12

Теперь мы можем использовать центральную колонку для создания разделителей.

Прежде всего, уберем отступы у всех блоков кроме middlebar. А для того, чтобы четко различать блоки добавим фоновый цвет для header и footer. В результате должно получиться что-то похожее на рис.13.

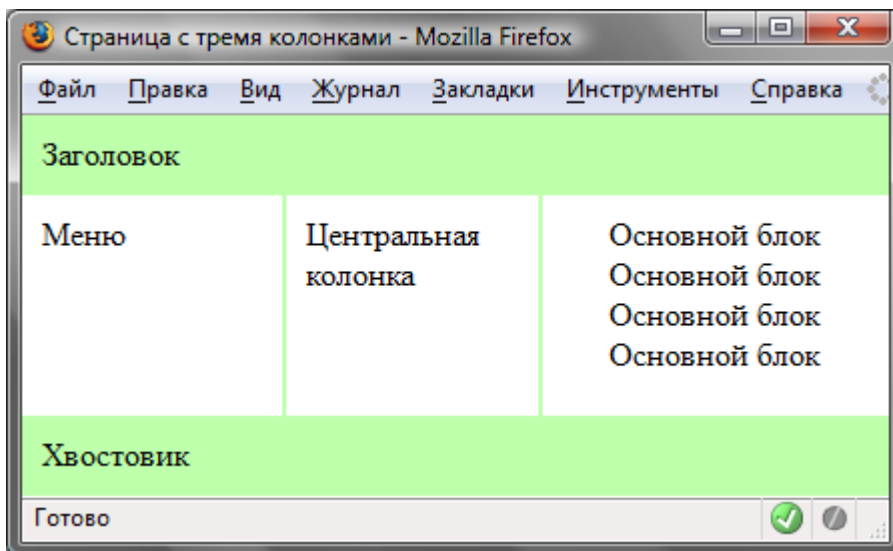


Рис.13



Заметьте. Эта страница отображается одинаково в трех основных браузерах (Firefox, Opera, IE).

Рассмотрим подробнее, какие изменения мы внесли в таблицу стилей.

```
01 body {
02   margin: 0;
03 }
04
05 #header {
06   margin: 0;
07   padding: 10px 10px 10px 10px;
08   background-color: #beffac;
09 }
10
```

```
11 #main {
12     margin: 0;
13     padding: 10px 10px 10px 10px;
14     float: right;
15     width: 30%;
16 }
17
18 #sidebar {
19     padding: 10px 10px 10px 10px;
20     margin: 0;
21     float: left;
22     width: 20%;
23 }
24
25 #middlebar {
26     border-left: solid 2px #beffac;
27     border-right: solid 2px #beffac;
28     margin: 0px 40% 0px 30%;
29     padding: 10px 10px 10px 10px;
30 }
31
32 #mfooter {
33     clear: both;
34 }
35
36 #footer {
37     margin: 0;
38     padding: 10px 10px 10px 10px;
39     background-color: #beffac;
40 }
```

Листинг 13

Прежде всего, мы убрали отступы у тега `body` (строки 1-3), что позволило разместить наши блоки вплотную к краям окна браузера.

Для всех блоков, кроме `middlebar` для свойства `margin` установлено значение 0. И убраны границы (свойство `border`).

Центральной колонке мы добавили левую и правую границы толщиной 2px (строки 26, 27).

Кстати, обратите внимание, мы убрали правило `clear: both` у блока `footer`, т.к. он в любом случае будет расположен ниже блока `mfooter`.

Как видите, ничего сложного. И, самое главное, мы получили нужный эффект не изменяя html разметку.



Вы можете использовать такой же способ для создания разделителя на странице с двумя колонками. Главное помните, что все блоки со свойством `float` должны находиться **до** блока, который вы растягиваете (имеется в виду расположение блоков в html разметке, в браузере они будут расположены рядом).

Заключение

В этом руководстве показан только один из возможных вариантов создания многоколоночных страниц. Каскадные таблицы стилей (CSS) позволяют решить одну и ту же задачу несколькими способами. И, вполне возможно, что вы найдете более подходящий вариант для вашего проекта.

Главное четко поставьте задачу, немного поэкспериментируйте, и получите хороший результат.

Удачи!

Полезные ссылки

www.etutorials.org.ua – здесь вы можете скачать все приведенные в руководстве примеры

www.simplecoding.org – блог автора этого руководства

www.csszengarden.com – лучший сайт, демонстрирующий возможности CSS и блочной верстки

<http://www.w3.org/> – основной источник информации о CSS и HTML

<http://www.css3.info/> – информация о CSS3

<http://jigsaw.w3.org/css-validator/> – online служба проверки каскадных таблиц стилей на соответствие стандартам

<http://gallery.theopalgroupp.com/selectoracle/> – эта служба создает текстовое описание CSS селекторов

<http://validator.w3.org/> – проверка html разметки

<http://www.positioniseverything.net/> – очень хороший сайт о CSS

<http://www.w3schools.com/> – что-то вроде справочника по web технологиям с хорошими примерами