

Блочная верстка web-сайта. Часть 1

1. Основные понятия

Многие web-дизайнеры отдают предпочтение **блочной верстке сайта**, которая осуществляется с помощью тега `<div>`. Страницы, сверстанные с помощью блочной верстки, значительно меньше весят по сравнению с табличной версткой.

Блок – это обычная прямоугольная область, обладающая рядом свойств, таких как: **рамка, поля и отступы** (рис. 1). Содержимым блока может быть что угодно – текст, картинки, списки, формы для заполнения, меню навигации и т.п.

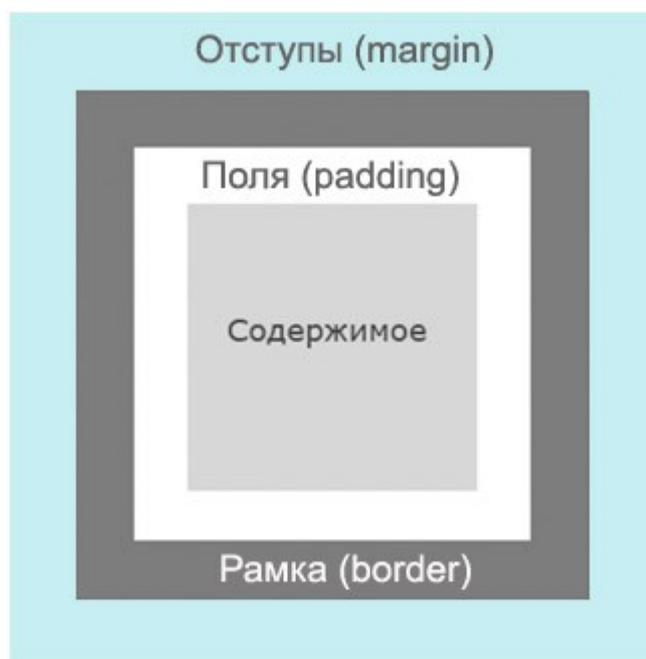


Рис. 1

Рамка (border) – это контур, для которого можно задать такие характеристики как толщина, цвет и тип (пунктирная, сплошная, точечная).

Поля (padding) – отделяют содержимое блока от его рамки, чтобы текст, например, не был «впритык» к стенкам блока.

Отступы (margin) – это пустое пространство между различными блоками, позволяющее на заданном расстоянии расположить два блока относительно друг друга.

Блоки, как и таблицы – это элементы, всегда располагающиеся на странице вертикально. То есть, если в коде страницы записаны подряд два блока, то отобразятся они в браузере один под другим. Если нам нужно расположить несколько блоков горизонтально, то в их свойствах задается такой параметр как «обтекание» (float). Но об этом чуть позже.

В данной работе мы создадим web-страничку из блоков. Сначала создадим контейнер, в который, как в коробку сложим наши блоки. Для наглядности каждый блок будет иметь свой цвет. Конечный результат должен быть таким как на рис. 2.

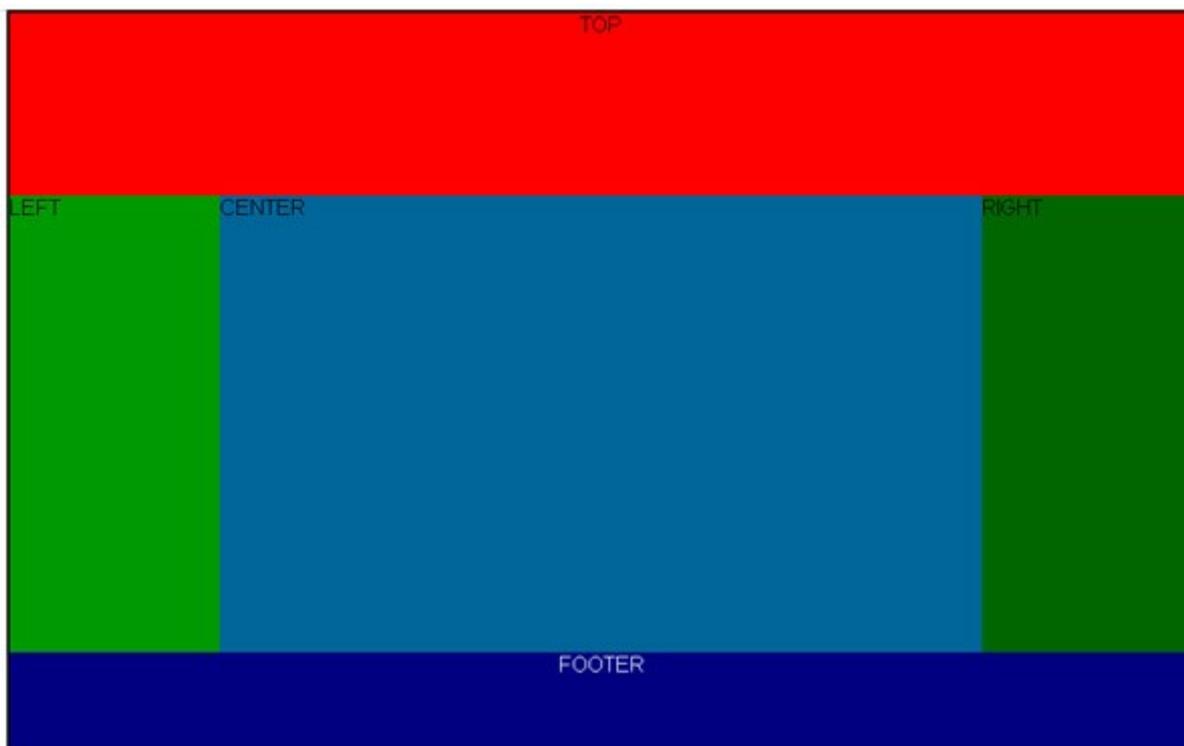


Рис. 2

Контейнер будет содержать в себе пять блоков:

TOP – шапка сайта, обычно содержит логотип компании, название, заголовки и слоганы, поиск, навигацию;

LEFT и RIGHT – левая и правая колонки, обычно содержат рекламу, навигацию, рассылку, новости и т. д.:

CENTER – содержит основной текст страницы;

FOOTER – подвал сайта, содержащий копирайты и другие сведения о сайте, возможно, дублировать навигацию, что является признаком хорошего тона.

Для теста нашего сайта нам понадобятся как минимум три самых популярных браузера – Opera, Fire Fox, Internet Explorer.

Описание web-страницы в основном делается в CSS документе.

2. «Фиксированный» дизайн методом блочной верстки

Задание 1

1. Создайте в блокноте новый документ с расширением css и сохраните его под именем **mystyle.css**.
2. Создайте HTML-документ и сохраните его в той же папке.

3. В самом начале HTML-документа впишите следующую строку: **<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">**

DOCTYPE

Определяет тип нашего документа. Любая грамотно сверстанная страница должна в самом начале содержать так называемый DOCTYPE. Нужен он для всевозможных устройств вывода информации и браузеров в том числе. Пока что все ныне существующие браузеры прекрасно обходятся и без указания DOCTYPE. Но уже грядут те времена, когда страница сайта, сразу начинающаяся с тэга `<html>`, просто не будет прочитана, потому как стандарты становятся жестче.

Мы определили нашу страничку в соответствии с DOCTYPE под названием **Strict 1.0**.

Требования здесь весьма строгие – все тэги, не имеющие закрывающей пары, должны заканчиваться пробелом со слэшем / перед закрывающей угловой скобкой. Но вот ведь сам DOCTYPE тоже выглядит как тэг! Почему же у него нет этого пробела со слэшем? А просто! Захотелось так разработчикам этих строгих правил. Но это единственный случай, где правило не работает.

4. Одной строкой между тэгами **<head>** и **</ head >** присоедините документ **mystyle.css** к документу HTML (рисунок 3).

```
<link href="mystyle.css" rel="stylesheet" type="text/css" />
```

Рисунок 3

5. В таблице стилей наберите код как на рисунке 4.

```
/* задаем параметры всего документа */
*{
  margin: 0; /* отступы 0 */
  padding: 0; /* поля 0 */
  border: 0; /* рамка 0 */
}

body {
  font-family: "Arial", Courier, monospace; /* шрифт документа */
  color: #000000; /* цвет шрифта */
  padding: 0; /* поля 0 */
  background: #FFFFFF; /* фон документа белый */
}

#container { /* создаем контейнер */
  width: 900px; /* ширина контейнера */
  margin: 0 auto; /* отступы сверху и снизу 0 справа и слева
автоматически */
  border: 2px #000 solid; /* рамка 2 пикселя черного цвета
сплошная */
}
```

Рисунок 4

этот знак говорит, что элемент является уникальным атрибутом и используется в HTML документе в теге **div** один раз.

6. Добавьте в **mystyle.css** шапку сайта (рисунок 5).

```
#top {
  width: 900px; /* ширина шапки */
  height: 140px; /* высота шапки */
  background: #FF0000; /* фоновый цвет */
  text-align: center; /*Выравнивание текста по центру блока*/
}
```

Рисунок 5

7. Добавим HTML документ следующий код между тегами **body** (рисунок 6).

```
<div id=" container">
  <div id=" top">TOP</div>
</div>
```

Рисунок 6

И у Вас должно получиться следующее (рис. 7).



Рисунок 7

Следующим этапом верстки сайта является расположение трех блоков последовательно по горизонтали, для этого в блочной верстке, как правило, используется элемент **float**. Он позволит обтекать другим элементам наш блок справа или слева.

8. Откройте css-документ и добавьте следующий код (рисунок 8).

```

#left_bar {
background:#009900; /* фон блока выбран условно */
width:160px; /* ширина блока */
height:350px; /* высота блока выбрана условно */
float:left; /* обтекание других блоков и элементов слева */
}

#main {
position: relative;
background:#006699;
width:580px;
height:350px;
float:left;
}

#right_bar {
background: #006600;
width:160px;
height:350px;
float:right; /* обтекание других блоков и элементов справа */
}

.clear {
clear: both; /*запрет обтекания с обеих сторон*/
}

```

Рисунок 8

Каждая строка кода имеет комментарий, заключенный в скобки /* комментарий */, который не отображается в браузере. Напомню, элементы со знаком # используются в теге **div id="main"** HTML-документа один только раз! Элемент **clear** содержит точку, которая идентифицирует его как **class**, может использоваться многократно, что и является различием между этими элементами. Теперь отразим все это в нашем HTML-документе.

9. Сразу после закрывающегося тега **<div id="TOP"></div>** вставим следующий код (рисунок 9).

```

<div id="left_bar">LEFT</div>
<div id="main">CENTER</div>
<div id="right_bar">RIGHT</div>
<div class="clear"></div>

```

Рисунок 9

10. Откройте HTML-документ в браузере. Должно получиться следующее (рисунок 10).



Рисунок 10

11. Теперь добавьте блок **footer** самостоятельно. Браузер должен показать следующее (рисунок 11).

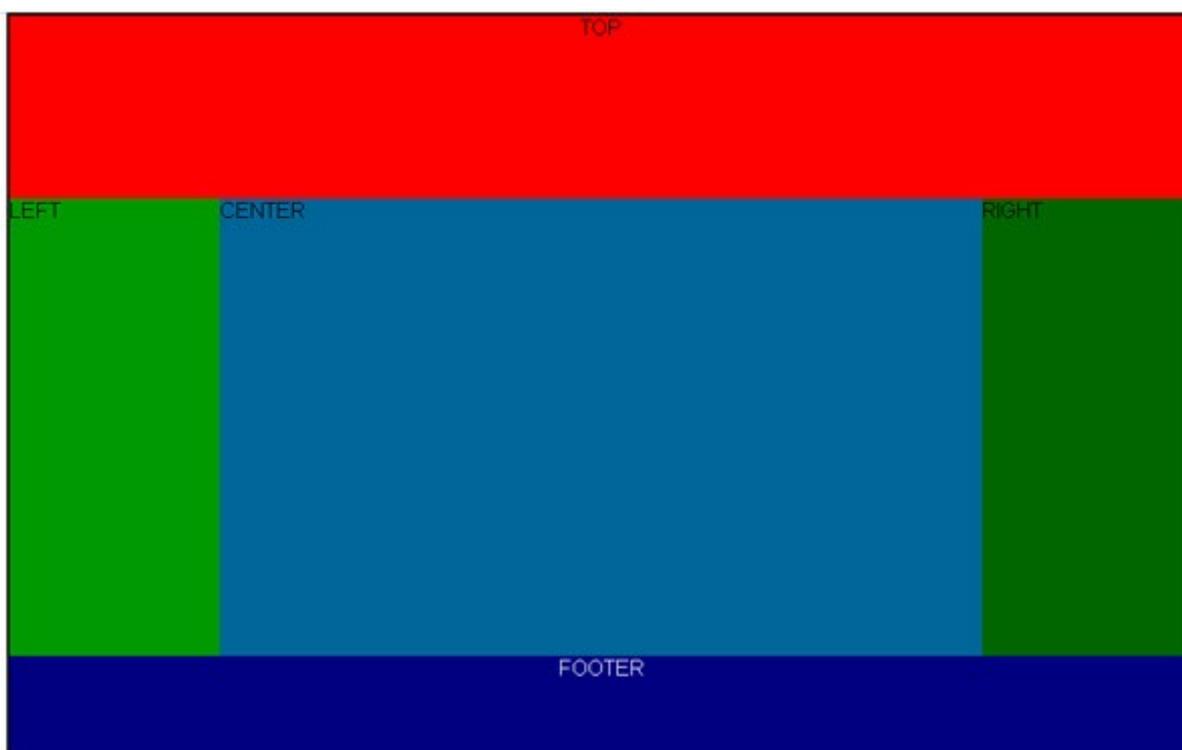


Рисунок 11

Задание 2

Рассмотрим атрибуты **relative** и **absolute**.

Иногда бывает необходимо разместить какой-то блок в строго заданном положении относительно родительского.

Рассмотрим простейший код.

1. Создайте html-документ, в теле которого разместите код, как на рисунке 12.

```
<div id="parent">
  <div id="child">«Это белый квадрат - дочерний элемент для
  черного квадрата»</div>
</div>
```

Рисунок 12

Предположим, мы хотим разместить дочерний блок в 20 пикселях левее и ниже верхнего левого угла родительского блока.

Попробуем сделать так (т.е. абсолютно позиционируем дочерний блок относительно родительского).

2. Создайте таблицу стилей, в которой наберите код из листинга на рисунке 13.

```
#parent {
  margin: 50px 100px;
  background: #000;
  width: 500px;
  height: 500px;
}

#child {
  position: absolute;
  top: 20px;
  left: 20px;
  width: 200px;
  height: 200px;
  background: #fff;
  border: 1px solid #000;
}
```

Рисунок 13

3. Проверьте web-страничку в браузере. Вот что получилось (рисунок 14). Это не то, что мы хотим, не так ли? Наш дочерний элемент ушел не к маме, а к дедушке (т.е. BODY)!

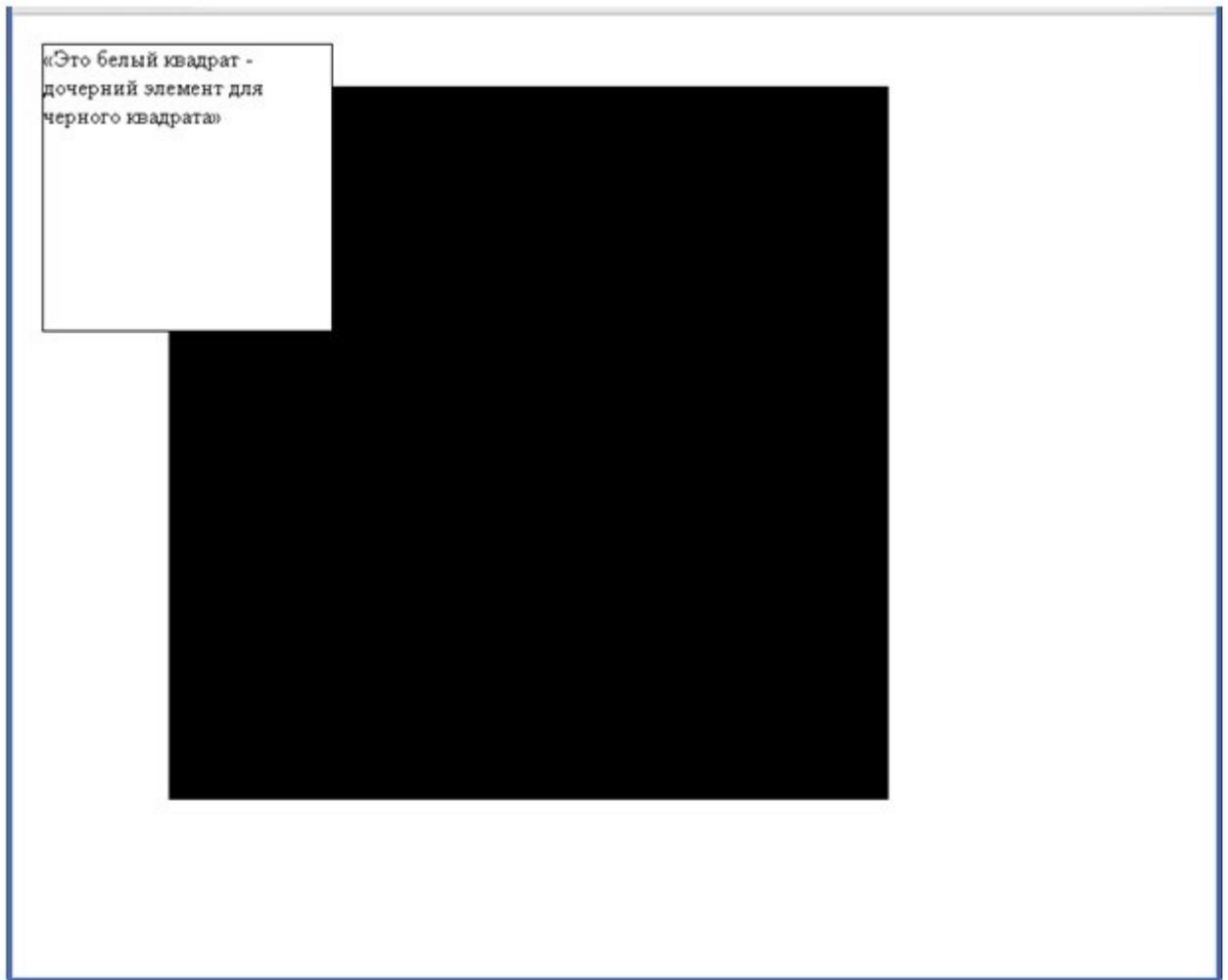


Рисунок 14

Проблема решается довольно просто: родителю дополнительно задаётся **position: relative;**

4. Измените код своей таблицы стилей в соответствии с рисунком 15.

```
#parent {  
  position: relative;  
  margin: 50px 100px;  
  background: #000;  
  width: 500px;  
  height: 500px;  
}  
  
#child {  
  position: absolute;  
  top: 20px;  
  left: 20px;  
  width: 200px;  
  height: 200px;  
  background: #fff;  
  border: 1px solid #000;  
}
```

Рисунок 15

5. Проверьте web-страничку в браузере. Результат на рисунке 16. Оцените разницу.

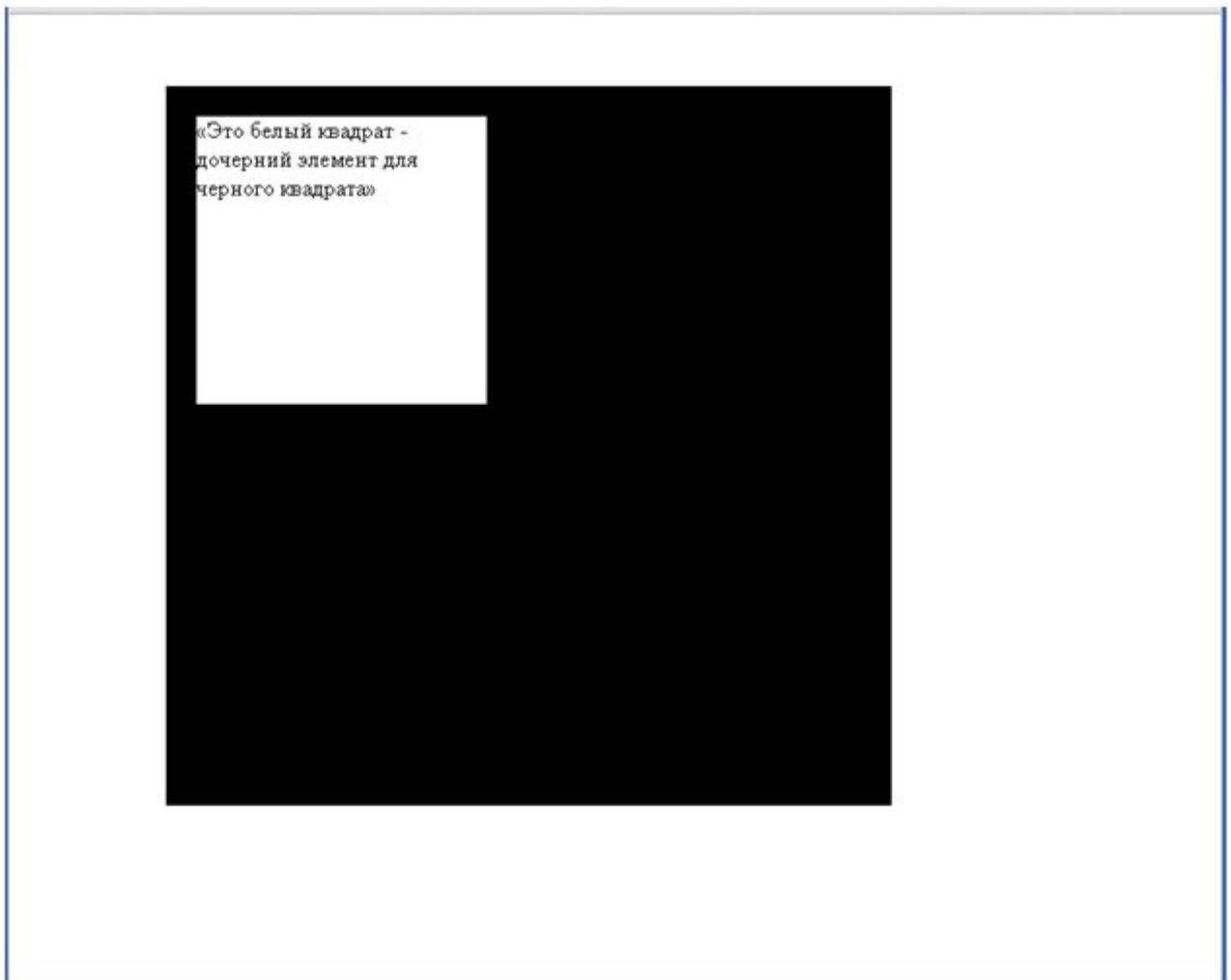


Рисунок 16

Контрольное задание

Создать web-страницу, внешний вид которой изображен ниже на рисунке 17



Рисунок 17

3. «Резиновый» сайт методом блочной верстки

В заданиях 1 и 2 мы рассмотрели «фиксированный» дизайн методом блочной верстки, т.к. все блоки имели точное значение по ширине и высоте в пикселах.

В случае «резинового» дизайна размеры блоков задаются в процентах от ширины экрана. По высоте размер блока может определяться содержимым блока.

Задание 3. Создадим «резиновый» блочный макет как на рисунке 18.

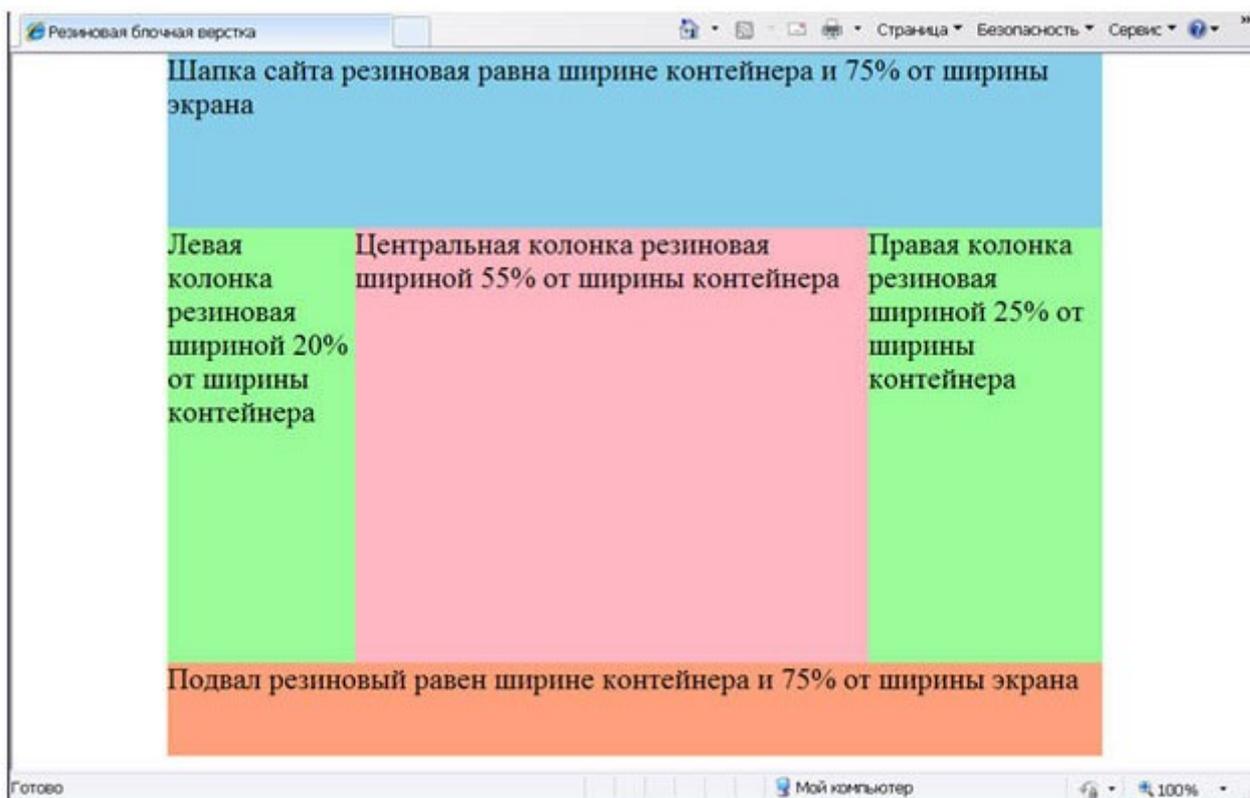


Рисунок 18

1. Создайте HTML-документ с кодом, как на рисунке 19. Сохраните документ.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html;
      charset=windows-1251" />
    <link href="blochn_6.css" rel="stylesheet" type="text/css" />
    <title>Резиновая блочная верстка</title>
  </head>
  <body>
    <div id="container">
      <div id="top">Шапка сайта резиновая равна ширине контейнера и 75% от
        ширины экрана</div>
      <div id="left_bar">Левая колонка резиновая шириной 20% от ширины
        контейнера</div>
      <div id="main">Центральная колонка резиновая шириной 55% от ширины
        контейнера</div>
      <div id="right_bar">Правая колонка резиновая шириной 25% от ширины
        контейнера</div>
      <div id="footer">Подвал резиновый равен ширине контейнера и 75% от
        ширины экрана</div>
    </div>
  </body>
</html>
```

Рисунок 19

2. Создайте таблицу стилей как на рисунке 20. Сохраните документ. Вы должны получить результат как на рисунке 18.
3. Изменяя размеры браузера, проанализируйте поведение макета.

```
1  *{                /* параметры всего документа */
2  margin: 0;        /* отступы 0 */
3  padding: 0;       /* поля 0 */
4  font-size:18pt;  /*высота шрифта на странице*/
5  }
6
7  body {            /*параметры тела документа*/
8  padding:0;        /* поля 0 */
9  }
10
11 #container {      /* создаем контейнер */
12 width:75%;         /* ширина контейнера в процентах от ширины экрана */
13 margin: 0 auto;   /* отступы сверху и снизу 0 справа и слева
14                    автоматически */
15 }
16
17 #top {             /*шапка сайта*/
18 width:100%;        /* ширина шапки  равна ширине контейнера*/
19 height:140px;     /* высота шапки */
20 background:#87CEEB;/* фоновый цвет  шапки*/
21 }
22
23 #left_bar {        /*левая колонка*/
24 width:20%;         /* ширина левой колонки  в процентах от ширины
25                    контейнера*/
26 height:350px;     /* высота левой колонки*/
27 background:#98FB98;/* фон левой колонки */
28 float:left;       /* обтекание других блоков слева */
29 }
30
31 #main {            /*центральная колонка*/
32 width:55%;         /*ширина центральной колонки в процентах от ширины
33                    контейнера*/
34 height:350px;     /*высота центральной колонки*/
35 background:#FFB6C1;/*фон центральной колонки*/
36 float:left;       /*обтекание других блоков слева*/
37 }
38
39 #right_bar {       /*правая колонка*/
40 width:25%;         /*ширина правой колонки в процентах от ширины
41                    контейнера*/
42 height:350px;     /*высота правой колонки*/
43 background:#98FB98;/*фон правой колонки*/
44 float:right;      /* обтекание других блоков справа */
45 }
46
47 #footer {          /*подвал сайта*/
```

Рисунок 20

4. Комбинированная блочная верстка

Комбинированная блочная верстка включает как блоки фиксированной ширины, так и блоки в процентном отношении к ширине экрана.

Задание 4. Создадим комбинированный блочный макет как на рисунке 21.

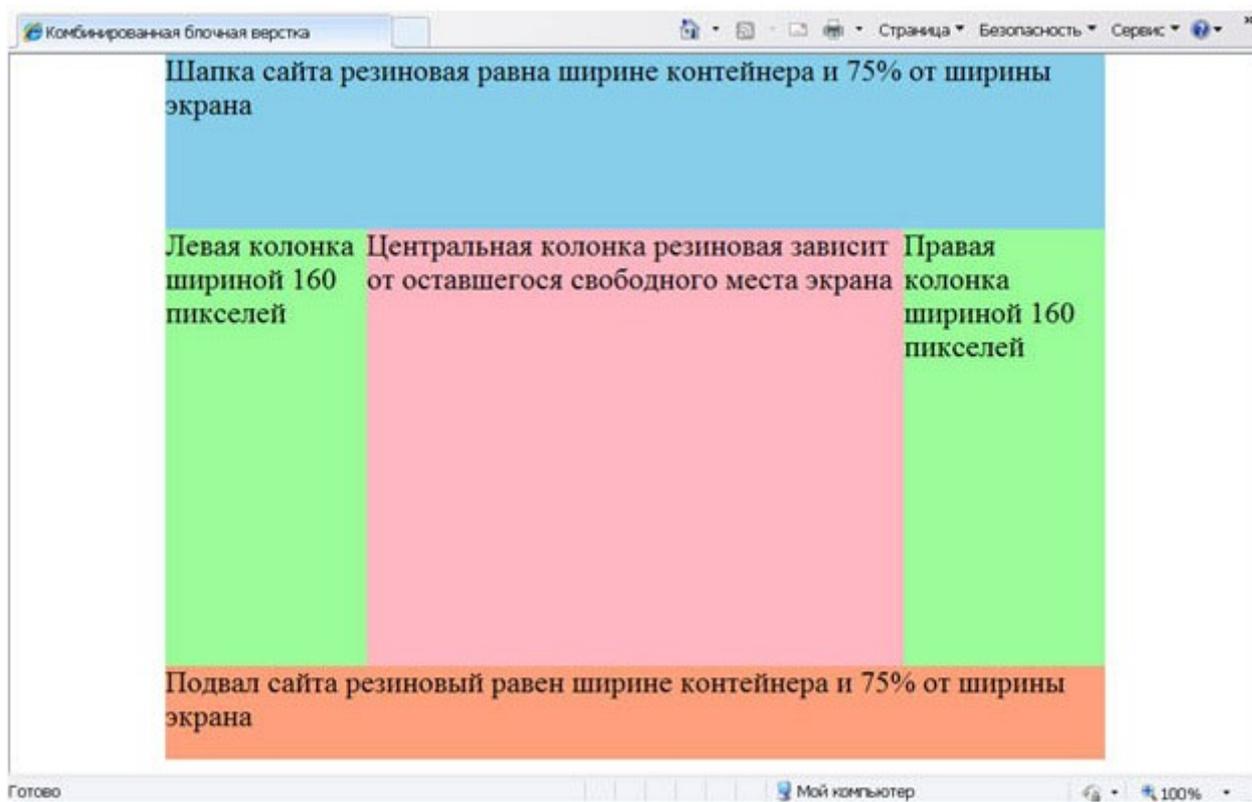


Рисунок 21

1. Создайте HTML-документ с кодом, как на рисунке 22.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html;
      charset=windows-1251" />
    <link href="blochn_7.css" rel="stylesheet" type="text/css" />
    <title>Комбинированная блочная верстка</title>
  </head>
  <body>
    <div id="container">
      <div id="top">Шапка сайта резиновая равна ширине контейнера и 75%
        от ширины экрана</div>
      <div id="left_bar">Левая колонка шириной 160 пикселей</div>
      <div id="main">Центральная колонка резиновая зависит от
        оставшегося свободного места экрана</div>
      <div id="right_bar">Правая колонка шириной 160 пикселей</div>
      <div id="footer">Подвал сайта резиновый равен ширине контейнера и
        75% от ширины экрана</div>
    </div>
  </body>
</html>
```

Рисунок 22

2. Создайте таблицу стилей как на рисунке 23. Сохраните документ. Вы должны получить результат как на рисунке 21.
3. Изменяя размеры браузера, проанализируйте поведение макета.

```

1  □*{          /* параметры всего документа */
2  |margin: 0;    /* отступы 0 */
3  |padding: 0;   /* поля 0 */
4  |font-size:18pt; /*высота шрифта на странице*/
5  |}
6
7  □body {      /*параметры тела документа*/
8  |padding:0;   /* поля 0 */
9  |}
10
11 □#container { /* создаем контейнер */
12 |width:75%;    /* ширина контейнера в процентах от ширины экрана */
13 |margin: 0 auto; /* отступы сверху и снизу 0 справа и слева
14 |автоматически */
15 |position:relative; /*объявляем контейнер родительским элементом*/
16 |}
17
18 □#top {      /*шапка сайта*/
19 |width:100%;   /* ширина шапки  равна ширине контейнера*/
20 |height:140px; /* высота шапки */
21 |background:#87CEEB; /* фоновый цвет  шапки*/
22 |}
23
24 □#left_bar { /*левая колонка*/
25 |width:160px;  /* ширина левой колонки */
26 |height:350px; /* высота левой колонки*/
27 |background:#98FB98; /* фон левой колонки */
28 |float:left;  /* обтекание других блоков слева */
29 |}
30
31 □#main {     /*центральная колонка*/
32 |           /*ширина центральной колонки не указываем вообще*/
33 |height:350px; /*высота центральной колонки*/
34 |background:#FFB6C1; /*фон центральной колонки*/
35 |margin-left:160px; /*отступ блока от левого края контейнера*/
36 |margin-right:160px; /*отступ блока от правого края контейнера*/
37 |}
38
39 □#right_bar { /*правая колонка*/
40 |width:160px;  /*ширина правой колонки*/
41 |height:350px; /*высота правой колонки*/
42 |background:#98FB98; /*фон правой колонки*/
43 |position:absolute; /*блок является дочерним элементов по отношению
44 |к контейнеру*/
44 |top:140px; /*отступ от верхнего края родительского элемента -
45 |контейнера*/
45 |right:0px; /*отступ от правого края родительского элемента -

```

Контрольное задание

Методом блочной верстки создайте web-страницу для сайта архитектурных проектов коттеджей так, как изображено на рис. 24. Изображение для шапки сайта (**shapka_div.jpg**).

Требования к макету:

- В HTML-коде не должно быть локального форматирования элементов, все оформление должно быть сделано через внешнюю таблицу стилей;
- При наведении на ссылку «Проекты таунхаусов и заблокированных домов» гиперссылка должна менять цвет (рис. 24);
- При нажатии на гиперссылку «Проекты таунхаусов и заблокированных домов» должна загружаться страница о проектах, внешний вид которой изображен на рис. 25.
- При нажатии на логотип должна загружаться главная страница сайта (рис. 24).



MyHouse.ru
КАТАЛОГ ПРОЕКТОВ

Каталог архитектурных проектов

ПРОЕКТЫ ДЛЯ ВАШЕГО БУДУЩЕГО ДОМА

Каталог готовых архитектурных проектов коттеджей myhouse.ru явился результатом многолетней работы лучших российских специалистов в области архитектуры и строительства. Это своего рода «Собрание сочинений» от современной загородной архитектуры, в котором каждый может найти что-то свое. Почти все проекты домов уникальны и лишь небольшое число коттеджей проектировались, как типовые; в широком же смысле слова, проекты коттеджей из каталога нельзя называть типовыми, поскольку разрабатывались они в порядке индивидуального проектирования непосредственно под нужды будущих владельцев загородных домов. Более 95% проектов реализовано в строительстве. Большая часть проектов реализована неоднократно, как в Московской области, так и по всей территории Российской Федерации, в странах ближнего зарубежья.

Заказать проект

Категории проектов

1. Проекты таунхаусов и блокированных домов
2. Проекты домов на две семьи
3. Проекты узких домов
4. Проекты бань

Каталог архитектурных проектов

- Проекты кирпичных домов
- Проекты домов из газобетона и пенобетона
- Проекты домов из бруса и бревна
- Проекты каркасных домов

Площади домов

- до 150 м²
- от 150 до 250 м²
- от 250 до 400 м²
- от 400 м²

Садовые участки



Рис. 24



MyHouse.ru
КАТАЛОГ ПРОЕКТОВ

Проекты таунхаусов и блокированных домов

В данном разделе каталога проектов загородных домов размещена выборка архитектурных проектов таунхаусов, домов с возможностью пристройки, блокировки.

Заказать проект

Категории проектов

1. Проекты таунхаусов и блокированных домов
2. Проекты домов на две семьи
3. Проекты узких домов
4. Проекты бань

Каталог архитектурных проектов

- Проекты кирпичных домов
- Проекты домов из газобетона и пенобетона
- Проекты домов из бруса и бревна
- Проекты каркасных домов

Площади домов

- до 150 м²
- от 150 до 250 м²
- от 250 до 400 м²
- от 400 м²

Садовые участки



Проект D-152-1D

Архитектурный проект дома из бруса.
Небольшой, но очень уютный

Площадь: 154.5 м²
Стены: дерево



Проект H-173-1P

Дом двухэтажный,
с двускатной кровлей.

Площадь: 173.2 м²
Стены: пенобетон



Проект V-208-1P

Не очень большой, но очень уютный дом
для всей семьи, подходящий как для
города, так и для дачи...

Площадь: 207.8 м²
Стены: пенобетон



Проект R-280-1D

Деревянный проект из бруса
с гаражом на 1 а/м.

Площадь: 278 м²
Стены: дерево

Рис. 25