**Блочная верстка web-сайта. Часть 1**

Наиболее популярной является блочная верстка сайта или div верстка. Наш урок блочной верстки сайта поможет начинающим освоить основные приемы div верстки.

**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.

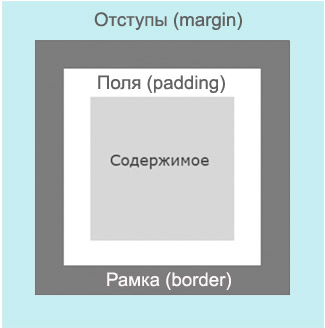


Рис. 1

**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.

Блочная верстка сайта включает в себя блоки. 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.

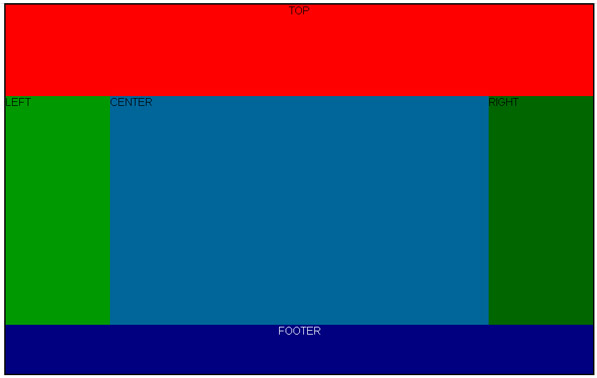


Рис. 2

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 тегами **<head> и </ head >** присоедините документ **mystyle.css** к документу HTML (рисунок 3).

http://web-verstka.ru/webverstka/kurs_html_css/images_html_css/7.3.jpg

Рисунок 3

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

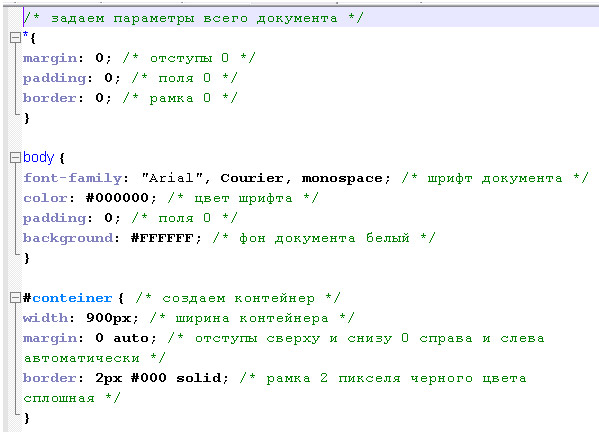


Рисунок 4

# 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).

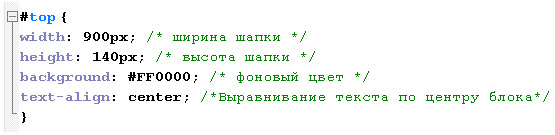


Рисунок 5

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

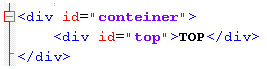


Рисунок 6

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



Рисунок 7

Следующим этапом блочной 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).

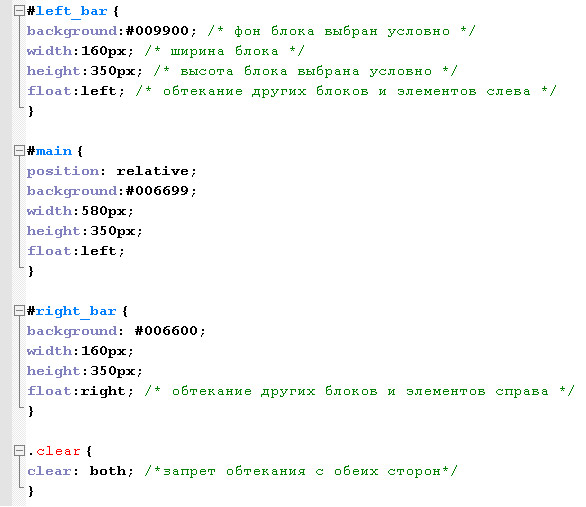


Рисунок 8

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).

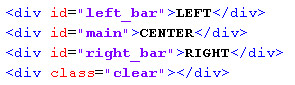


Рисунок 9

10. Откройте HTML-документ в браузере. Должно получиться такая div верстка (рисунок 10).

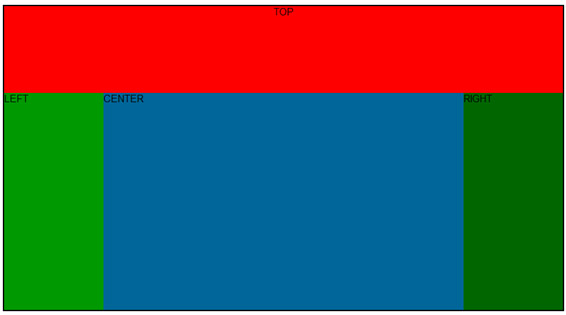


Рисунок 10

11. Теперь добавьте блок **footer** самостоятельно. Браузер должен показать такую блочную верстку сайта (рисунок 11).

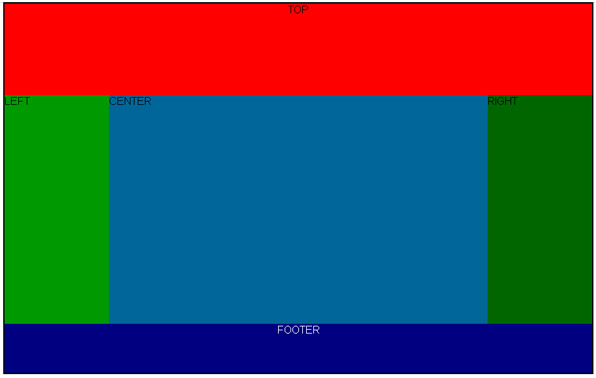


Рисунок 11

**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.

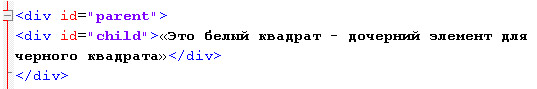


Рисунок 12

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.

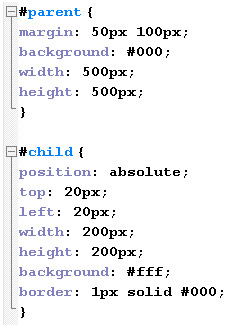


Рисунок 13

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)!

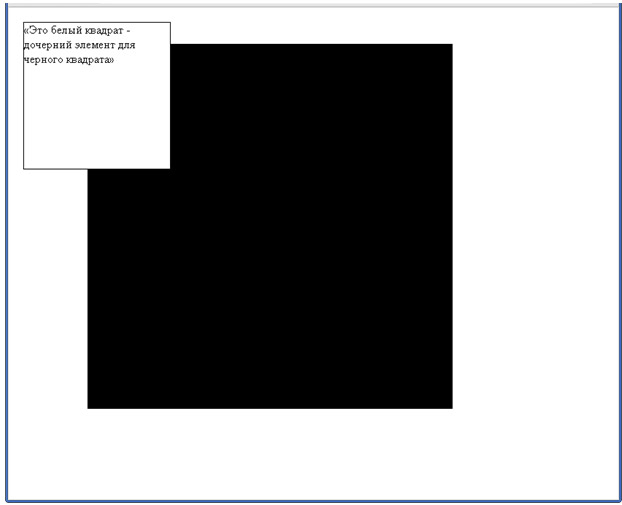


Рисунок 14

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

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

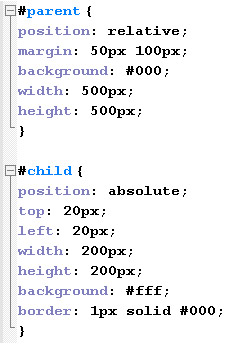


Рисунок 15

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

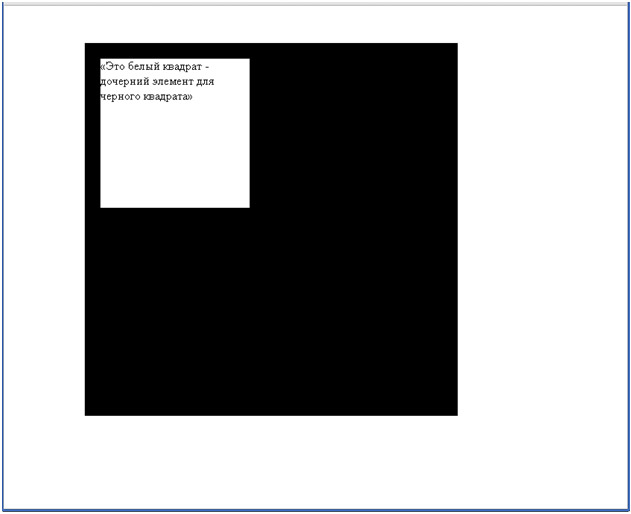


Рисунок 16

**Контрольное задание**

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

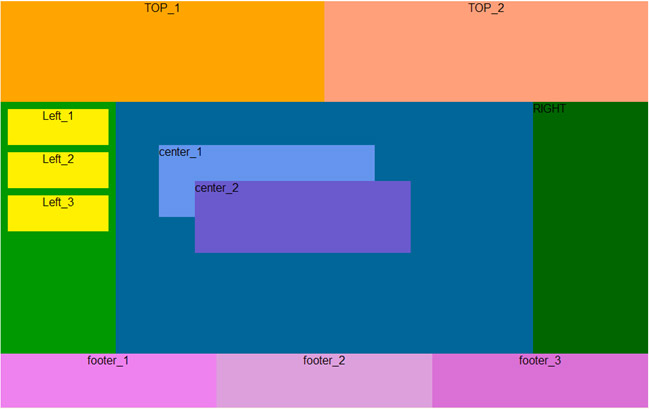


Рисунок 17

**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.

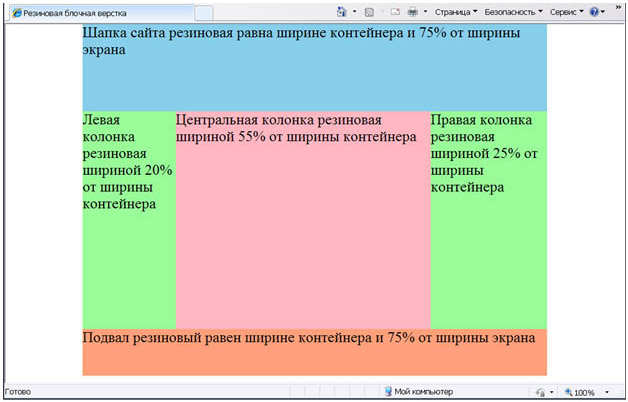


Рисунок 18

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

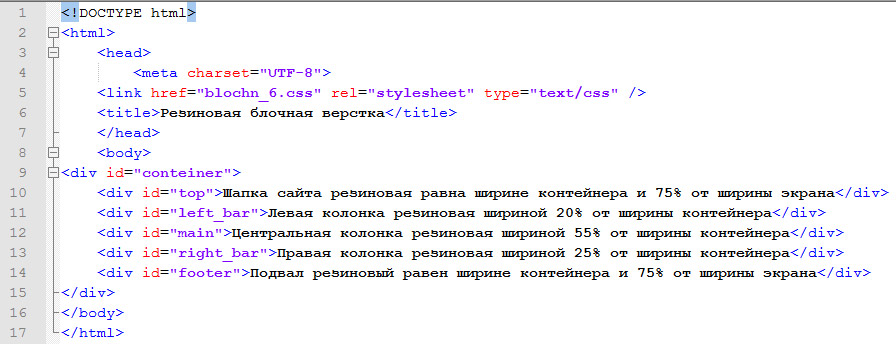


Рисунок 19

2. Создайте таблицу стилей как на рисунке 20. Сохраните документ. Вы должны получить результат ка на рисунке 18.

3. Изменяя размеры браузера, проанализируйте поведение макета.

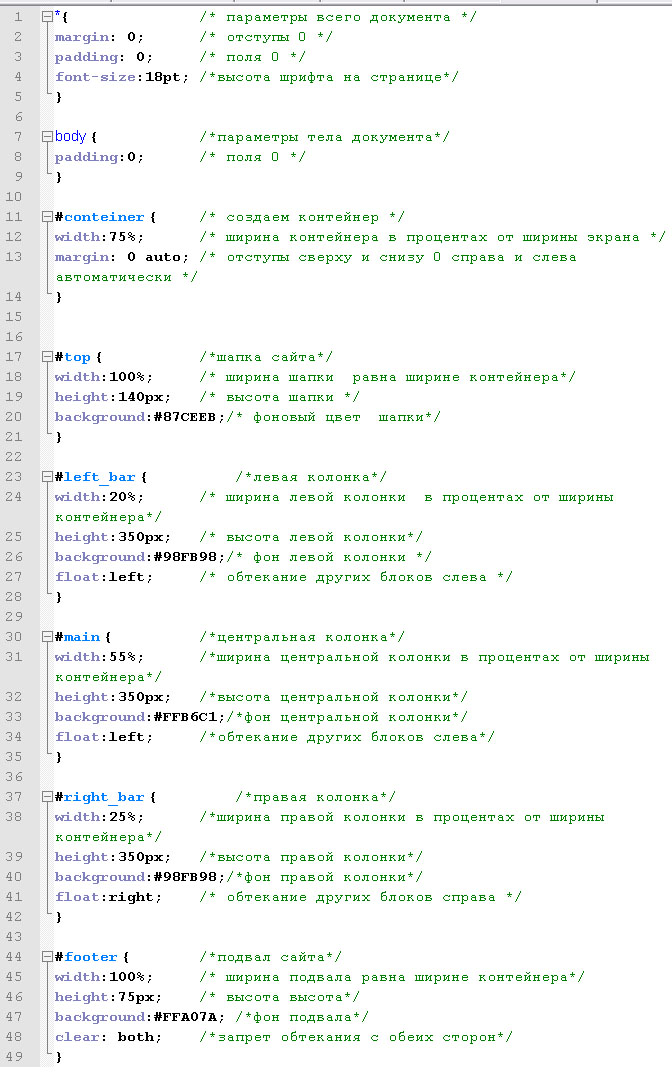


Рисунок 20

**4. Комбинированная блочная верстка**

Комбинированная блочная верстка (div 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.

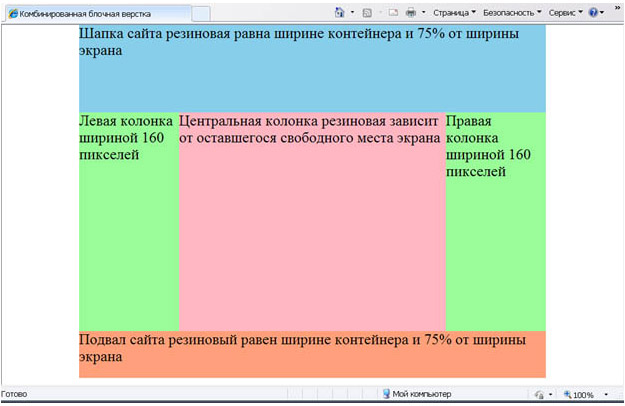


Рисунок 21

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

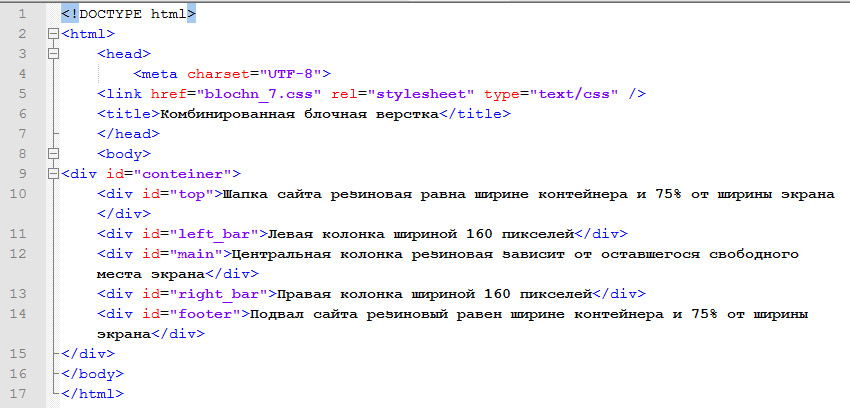


Рисунок 22

2. Создайте таблицу стилей как на рисунке 23. Сохраните документ.  Вы должны получить результат ка на рисунке 21.

3. Изменяя размеры браузера, проанализируйте поведение макета.



Рисунок 23

**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 блокированных домов» гиперссылка должна менять цвет (рис. 24);
* При нажатии на гиперссылку «Проекты таунхаусов и блокированных домов» должна загружаться страница о проектах, внешний вид которой изображен на рис. 25.
* При нажатии на логотип должна загружаться главная страница сайта (рис. 24).



Рис. 24



Рис. 25