

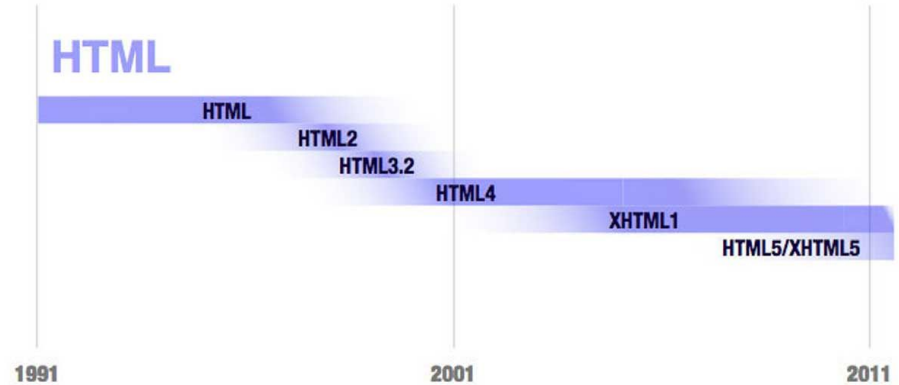
Блочная вёрстка

Стандарты HTML

Что за стандарты?

Когда-то давно четыре народа жили в мире...

Когда-то давно существовало множество неофициальных стандартов HTML, которые развивались отдельными лицами. Разработчикам браузеров было сложно принять всё во внимание. Именно поэтому в 1995 Инженерный совет интернета (IETF) выпустил единый стандарт HTML. Получил он версию 2.0 для того чтобы отличался от неофициальных стандартов. И с тех пор началось...

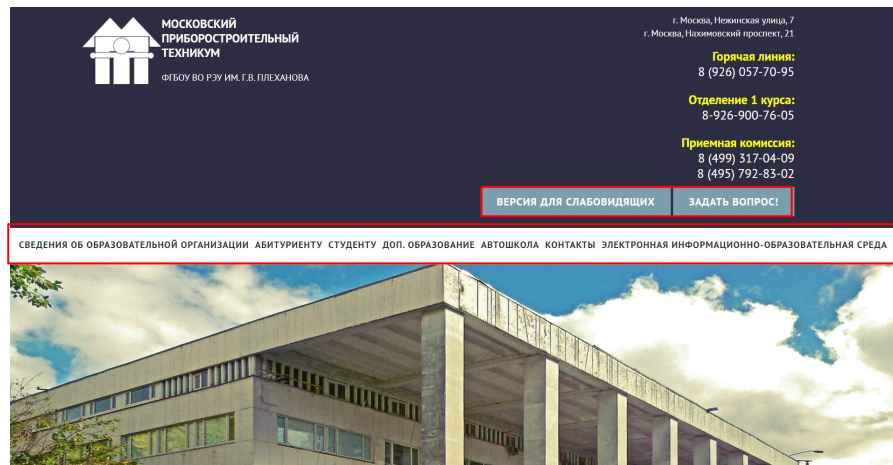


Немного истории

Проблема при вёрстке веб-страницы

Вы могли заметить, что различные теги, например, `<p>`, заголовков располагается друг под другом, а что если нужно расположить элементы в несколько столбцов? Использовать боковые меню и прочее?

На начальных этапах развития веба была проблема позиционирования элементов и текста на веб-странице

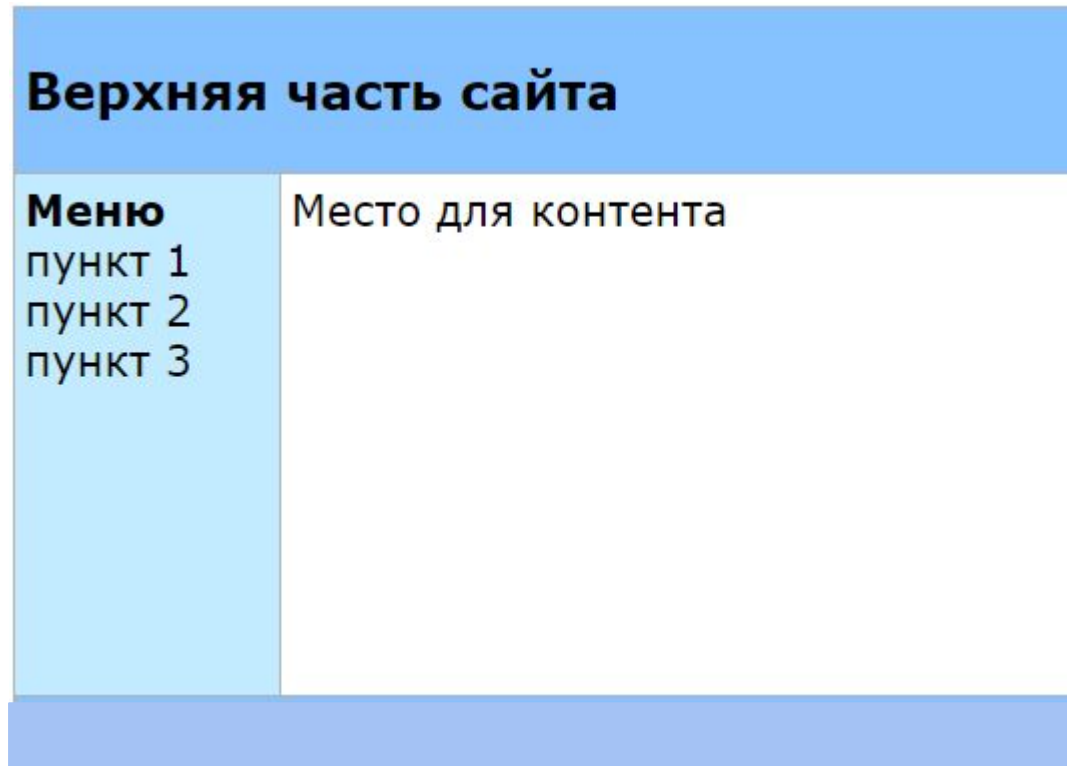


Табличная вёрстка

Табличная вёрстка

Решением проблемы была табличная вёрстка, которая была актуально в начале 2000-ых годах. Выглядела она следующим образом

```
1 <html>
2 <body>
3
4
5 <table style="width: 500px;border:0px;">
6 <tr>
7 <td colspan="2" style="background-color:#85C2FF;">
8 <h1 style="color: black;">Верхняя часть сайта</h1>
9 </td>
10 </tr>
11 <tr style="vertical-align: top;">
12 <td style="background-color:#C2EBFF;width:100px;text-align:top;">
13 <b>Меню</b><br>
14 пункт 1<br>
15 пункт 2<br>
16 пункт 3
17 </td>
18 <td style="background-color:#fff;height:200px;width:400px;text-align:top;">
19 Место для контента</td>
20 </tr>
21 <tr>
22 <td colspan="2" style="background-color:#85C2FF;text-align:center;">
23 Copyright © 2012 www.puzzleweb.ru</td>
24 </tr>
25 </table>
26
27 </body>
</html>
```



Достоинства и недостатки табличной вёрстки

Достоинства

- Резиновый макет. Табличный способ верстки позволяет легко сделать резиновый макет, задав ширину в процентах.
- Создание декоративных эффектов. Каждой ячейке можно задать отдельный фоновый рисунок и размеры.
- Вертикальное выравнивание. Внутри таблиц поддерживается выравнивание, как по горизонтали, так и по вертикали.
- Кроссбраузерность. Теги для табличных данных появились очень давно и поддерживаются даже в старейших версиях Internet Explorer. Сайт на таблицах одинаково выглядит в разных браузерах.

Недостатки

- Огромный код. Для вывода таблицы требуется множество тегов. В реальных примерах табличной верстке часто встречаются еще и вложенные таблицы. В-общем, даже на разметку достаточно простой страницы может понадобиться 200+ строчек кода.
- Ухудшение скорости загрузки. Таблица имеет одну очень нехорошую особенность – она не будет отображена до тех пор, пока все ее ячейки, включая самые нижние, не будут загружены.
- Плохая индексация. А вот это еще более значительный минус. Из-за большой разметки и обилия вложенных тегов текст на странице хуже воспринимается поисковыми системами
- Неудобное управление через стили. Опять же, из-за обилия вложенных тегов таблицу сложно стилизовать через стили.
- Несоответствие стандартам разработки.

Вывод по табличной вёрстке

Данный способ был актуален в бородатые времена когда альтернатив по сути небыло и приходилось мириться со всеми представленными недостатками. В данный момент такой способ разработки веб-страницы не используется и в 2011 году был признан неактуален.



Блочная вёрстка

Блочная вёрстка

Примерно в одно время с табличной вёрсткой был ещё один тип вёрстки, а именно блочная вёрстка.

```
<!DOCTYPE html>
<html>
<head>
  <title>Блочная вёрстка</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div id="container">
    <div id="header">
      <h2>header (шапка сайта)</h2>
    </div>

    <div id="navigation">
      <h2>Блок навигации</h2>
    </div>

    <div id="sidebar">
      <h2>Левая панель</h2>
    </div>

    <div id="content">
      <h2>Основной контент страницы</h2>
    </div>

    <div id="clear">
    </div>

    <div id="footer">
      <h2>footer (низ сайта)</h2>
    </div>
  </div>
</body>
</html>
```

```
body {
  background: #FFF;
  color: #000;
  font-family: Arial, sans-serif;
  font-size: 14px;
}

#header {
  background: #F5DEB3;
  width: 100%;
  height: 55px;
}

#container {
  background: #FFD700;
  margin: auto auto;
  text-align: center;
  width: 80%;
  height: 400px;
}

#navigation {
  background: #FE9798;
  width: 100%;
  height: 25px;
}

#sidebar {
  background: #40E0D0;
  float: left;
  width: 20%;
  height: 280px;
}

#content {
  background: #DCDCDC;
  float: right;
  width: 80%;
  height: 280px;
}

#clear {
  clear: both;
}

#footer {
  background: #00BFFF;
  width: 100%;
  height: 40px;
}
```



Достоинства и недостатки

Достоинства

- Меньший объем кода.
- Удобство изменения элементов. Изменить практически любой из элементов внешнего вида макета сайта можно путем правки файла стилей.
- Преимущества в плане SEO.
- Читабельность кода.
- Реализация сложных задач. С помощью блоков реализовать различные задачи по нестандартному расположению и оформлению элементов сайта гораздо проще, чем при использовании таблиц.
- Возможность создания адаптивного дизайна.

Недостатки

- Сложности в освоении. Если начать работать с таблицами может начать и новичок в сайтостроении, то верстку слоями освоить сложнее, управление блоками требует хороших знаний не только HTML, но и CSS.
- - Горизонтальное размещение элементов на веб-странице. Для того чтобы разместить несколько блоков в горизонтальной плоскости разработчику приходилось прибегать к различным хитростям(В HTML5 данная проблема была устранена).
- - Центрирование элементов по обеим осям

Вывод по блочной вёрстке

Хотя и блочная вёрстка более сложна в освоении, тем не менее преимущества перекрывают данный недостаток, поэтому в HTML4 была более предпочтителен именно данный способ создания веб-страниц. В HTML5 также была устранена сложность при горизонтальном расположении блоков.



Практика по блочной вёрстке

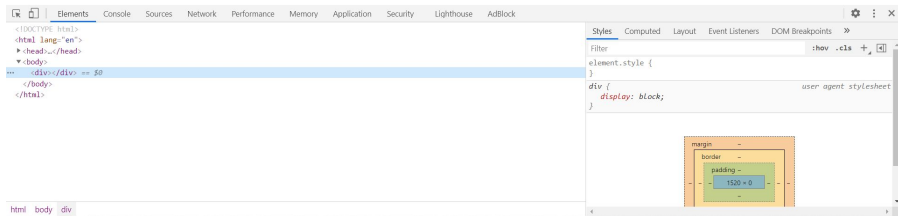
Что же это?

Блочная верстка — это подход, при котором сайт строят на основе «блоков», в качестве которых выступают, как правило, теги `<div>`. В англоязычных источниках такой подход называют Layouts. Блоки при таком подходе располагаются один под другим, но при помощи стандарта CSS разработчики могут менять их порядок отображения и позиционирование, а также задавать одному объекту различные стили на различных разрешениях браузеров. По сути это просто контейнер, который удобно стилизовать.

Как создать блок?

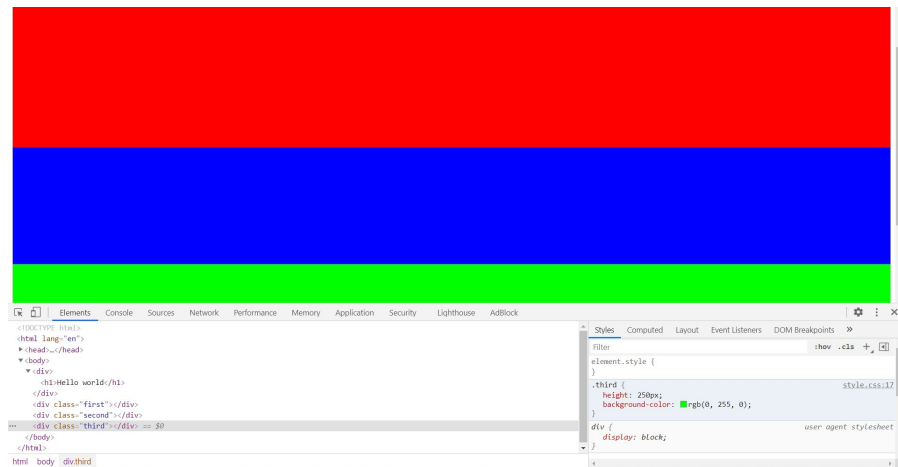
Для создания блока используется парный тег `<div>` elements `</div>`. Если создать блок без начинки, то он будет иметь значение высоты - 0px, а значение ширины - 100% родителя.

div 1520 x 0



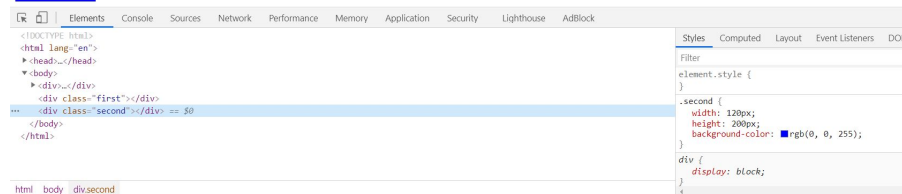
Обязательные условия

Все блочные элементы должны содержать в себе какой-либо селектор (селектор класса или селектор идентификатора). Также важно помнить, что каждый новый блок располагается под предыдущем. При этом html всё равно есть ли свободное пространство или нет. Каждый новый блок всегда будет ниже предыдущего.



Вертикальное выравнивание

Блоки не выравниваются горизонтально, даже если есть свободное пространство.



Блок в блоке ©Xzibit (сейчас будет сложно)

Блок может содержать не только другие элемент, но и другие блоки. Именно благодаря помещению блоков в блок в будущем мы научимся вырывать их по горизонтали, центровать и т.д. Также если раньше бы мы задали блоку процентную высоту, то он бы всё равно остался бы 0px. А теперь...

