

Лабораторная работа №2. Каскадные таблицы стилей. Блочная вёрстка.

На прошлой лабораторной работе вы, выполнив табличную вёрстку веб-страниц, наверное заметили, что страница представляет собой нагромождение тегов `table`, `tr`, `td`, в которых при чуть усложнённой структуре документа становится сложно ориентироваться. Плюс ко всему, табличная вёрстка – не семантическая. Таблицы должны использоваться только для табличных данных. В рамках этой лабораторной работы, вы должны переверстать свои страницы в блочном стиле и «раскрасить» их средствами CSS.

Условная классификация элементов HTML

Блочные элементы

Блочным называется элемент, который отображается на веб-странице в виде прямоугольника. Такой элемент занимает всю доступную ширину, высота элемента определяется его содержимым, и он всегда начинается с новой строки. К блочным элементам относятся теги

`<address>`, `<blockquote>`, `<div>`, `<fieldset>`, `<form>`, `<h1>`, ..., `<h6>`, `<hr>`, ``, `<p>`, `<pre>`, `<table>`, ``

и некоторые устаревшие. Также блочным становится элемент, если в стиле для него свойство `display` задано как `block`, `list-item`, `table` и в некоторых случаях `run-in`.

Для блочных элементов характерны следующие особенности.

- Блоки располагаются по вертикали друг под другом.
- На прилегающих сторонах элементов действует эффект схлопывания отступов.
- Запрещено вставлять блочный элемент внутрь строчного. Например, `<a><h1>Заголовок</h1>` не пройдёт валидацию, правильно вложить теги наоборот — `<h1><a>Заголовок</h1>`.
- По ширине блочные элементы занимают всё допустимое пространство.
- Если задана ширина контента (свойство `width`), то ширина блока складывается из значений `width`, полей, границ, отступов слева и справа.
- Высота блочного элемента вычисляется браузером автоматически, исходя из содержимого блока.
- Если задана высота контента (свойство `height`), то высота блока складывается из значения `height`, полей, границ, отступов сверху и снизу. При превышении указанной высоты контент отображается поверх блока.
- На блочные элементы не действуют свойства, предназначенные для строчных элементов, вроде `vertical-align`.
- Текст по умолчанию выравнивается по левому краю.

Строчные элементы

Строчными называются такие элементы документа, которые являются непосредственной частью строки. К строчным элементам относятся теги ``, ``, `<a>`, `<q>`, `<code>` и др., а также элементы, у которых свойство `display` установлено как `inline`. В основном они используются для изменения вида текста или его логического выделения.

По аналогии с блочными элементами перечислим их характерные особенности.

- Внутри строчных элементов допустимо помещать текст или другие строчные элементы. Вставлять блочные элементы внутрь строчных запрещено.
- Эффект схлопывания отступов не действует.
- Свойства, связанные с размерами (`width`, `height`) не применимы.

- Ширина равна содержимому плюс значения отступов, полей и границ.
- Несколько строчных элементов идущих подряд располагаются на одной строке и переносятся на другую строку при необходимости.
- Можно выравнивать по вертикали с помощью свойства `vertical-align`.

Строчные элементы удобно использовать для изменения вида и стиля текста, в частности, отдельных символов и слов. Для этой цели обычно применяется универсальный тег ``, который самостоятельно никак не модифицирует содержимое, но легко объединяется со стилями через классы или идентификаторы. За счёт чего с помощью этого тега можно легко управлять видом и положением отдельных фрагментов текста или рисунков.

Для вёрстки строчные элементы применяются реже, чем блочные. Это связано в основном с тем, что внутри строчных элементов не допускается вкладывать контейнеры `<div>`, `<p>` и подобные широко распространённые теги. Тем не менее, блочные и строчные элементы удачно дополняют друг друга, поскольку позволяют на всех уровнях менять вид составляющих веб-страниц.

Блочно-строчные элементы

Блочные и строчные элементы отлично дополняют друг друга при вёрстке, занимая каждый свою определённую нишу. Но возникают случаи, когда характеристик этих элементов явно недостаточно.

Галерея фотографий, представленная на рисунке 2.1 состоит из секций, в которые входит изображение с подписью к нему, при этом секции выстраиваются по горизонтали, занимая всю доступную ширину. При уменьшении окна браузера секции переходят на другую строку.

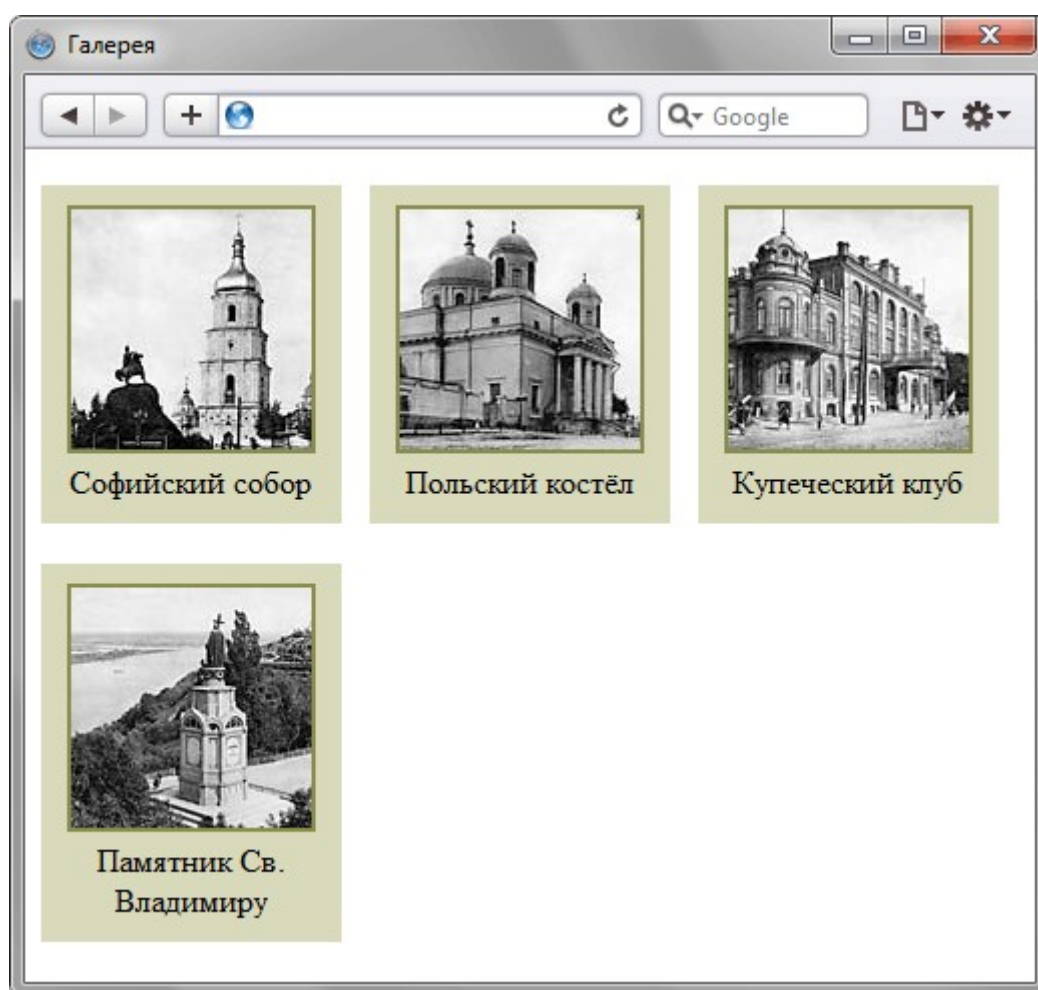


Рис. 2.1

Если для формирования секций использовать тег `<div>`, как блочный элемент он будет каждый раз начинаться с новой строки. Для строчных элементов нельзя задать цвет фона всей секции и установить её размеры. Наиболее популярное решение в подобных случаях это использование свойства `float`, которое будет рассмотрено в следующем разделе. Пока же остановимся на строчно-блочных элементах, которые сочетают преимущества строчных и блочных элементов.

В HTML нет тега, который относится к строчно-блочным элементам, его можно определить, задав элементу свойство `display` со значением `inline-block`.

```
div {  
  display: inline-block;  
}
```

Характеристики этих элементов следующие.

- Внутри строчно-блочных элементов допустимо помещать текст, строчные или блочные элементы.
- Высота элемента вычисляется браузером автоматически, исходя из содержимого блока.
- Ширина равна содержимому плюс значения отступов, полей и границ.
- Несколько элементов идущих подряд располагаются на одной строке и переносятся на другую строку при необходимости.
- Можно выравнивать по вертикали с помощью свойства `vertical-align`.
- Разрешено задавать ширину и высоту.
- Эффект схлопывания отступов не действует.

Пример такой вёрстки:

```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  <title>Галерея</title>  
  <style type="text/css">  
    .photo {  
      background: #d9dabb; /* Цвет фона */  
      width: 150px; /* Ширина */  
      margin: 0 10px 10px 0; /* Отступы */  
      padding: 10px 0; /* Поля сверху и снизу */  
      text-align: center; /* Выравнивание по центру */  
      display: inline-block; /* Строчно-блочный элемент */  
    }  
    .photo img {  
      border: 2px solid #8b8e4b; /* Параметры рамки */  
    }  
    .photo p {  
      margin: 0; /* Отступы */  
    }  
  </style>  
</head>  
<body>  
  <div class="photo">  
    <p>  
        
    </p>  
    <p class="caption">Софийский собор</p>  
  </div>  
  <div class="photo">  
    <p>  
        
    </p>  
  </div>  
</body>  
</html>
```

```

    <p class="caption">Польский костёл</p>
</div>
<div class="photo">
    <p>
        
    </p>
    <p class="caption">Купеческий клуб</p>
</div>
<div class="photo">
    <p>
        
    </p>
    <p class="caption">Памятник Св. Владимиру</p>
</div>
</body>
</html>

```

Плавающие элементы

Плавающими будем называть такие элементы, которые обтекаются по контуру другими объектами веб-страницы, например, текстом. Правильнее говорить «обтекаемые элементы», но с другой стороны термин «плавающий элемент» давно уже прижился, так что его и будем использовать в дальнейшем.

Плавающие элементы достаточно активно применяются при вёрстке веб-страниц и служат для реализации этих и не только задач:

- обтекание картинок текстом;
- создание врезок;
- горизонтальные меню;
- колонки.

Обтекание происходит с помощью стилевого свойства `float` со значением `left` или `right`. По умолчанию обтекание для элементов не устанавливается, но если это по каким-либо причинам необходимо указать явно, следует использовать значение `none`.

По умолчанию блочные элементы выстраиваются по вертикали один под другим, но при помощи свойства `float` их можно заставить располагаться рядом по горизонтали. При этом требуется установить ширину слоёв и задать для них `float`. Если ширина не указана, она будет равна содержимому слоя с учётом полей и границ.

Пример, рисунок 2.2:

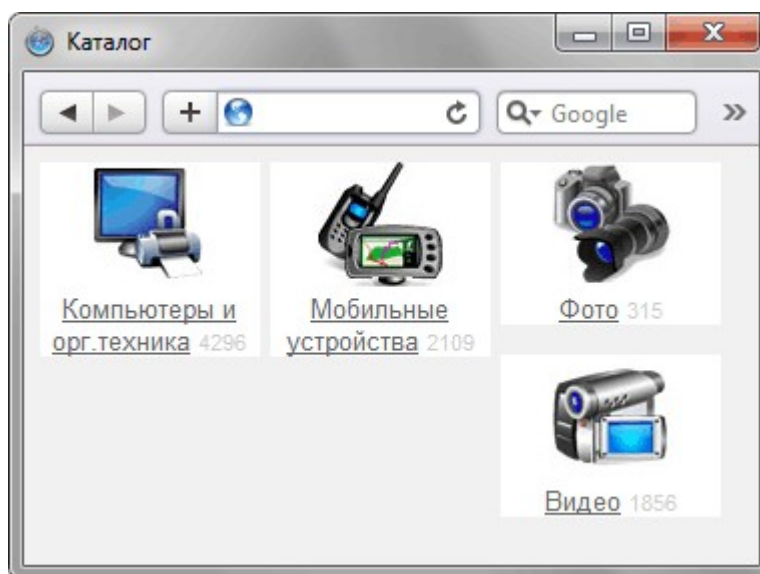


Рис. 2.2

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Каталог</title>
  <style type="text/css">
    BODY {
      font: 10pt Arial, Helvetica, sans-serif;
      background: #f0f0f0; /* Цвет фона веб-страницы */
    }
    #catalog A { color: #666; }
    #catalog A:hover { color: #1fa0e2; }
    #catalog DIV {
      width: 110px; /* Ширина */
      background: #fff;
      margin: 0 5px 15px 0; /* Отступы */
      text-align: center; /* Выравнивание по центру */
      float: left; /* Выравнивание по левому краю */
    }
    #catalog P { margin: 0 5px; }
    #catalog SPAN { color: #ccc; font-size: 0.8em; }
  </style>
</head>
<body>
  <div id="catalog">
    <div>
      <p>  </p>
      <p> <a href="#">Компьютеры и орг.техника</a> <span>4296</span> </p>
    </div>
    <div>
      <p>  </p>
      <p> <a href="#">Мобильные устройства</a> <span>2109</span> </p>
    </div>
    <div>
      <p>  </p>
      <p> <a href="#">Фото</a> <span>315</span> </p>
    </div>
    <div>
      <p>  </p>
      <p> <a href="#">Видео</a> <span>1856</span> </p>
    </div>
  </div>
</body>
</html>

```

Макеты блочной вёрстки

2-х колоночная вёрстка

Опять же, существует несколько подходов к формированию такого макета, но самый простой и удобный заключается в сочетании свойств `float` и `margin`.

Рассмотрим вариант, когда левая колонка имеет определённый размер, а ширина правой колонки устанавливается автоматически, исходя из ширины окна браузера. При этом ширина левой колонки может задаваться в пикселах или процентах. В таблице 2.1 приведены основные стилевые свойства для формирования двух колонок.

Для левой колонки требуется всего два свойства: `float` — заставляет вторую колонку располагаться рядом по горизонтали с первой и `width`, которое устанавливает ширину колонки. Вторая колонка будет занимать всё оставшееся место, поэтому для неё указывать `width` не нужно.

Правая колонка характеризуется лишь одним свойством — `margin-left`, оно смещает левый край колонки на ширину левого слоя, плюс задаёт отступ между колонками. Поэтому величина этого свойства в таблице 2.1 указана 21%, где 20% сама ширина колонки, а на один процент приходится расстояние между колонками. В случае задания ширины одной из колонок в пикселах, код останется прежним, но поменяются единицы измерения.

Табл. 2.1

Для левого слоя шириной 20%	
Левая колонка	Правая колонка
float: left width: 20%	margin-left: 21%
Для левого слоя шириной 200px	
float: left width: 200px	margin-left: 210px

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>...</title>
<style type="text/css">
    body {
        font: 10pt Arial, Helvetica, sans-serif; /* Шрифт на веб-странице */
        background: #e1dfb9; /* Цвет фона */
    }
    h2 {
        font-size: 1.1em; /* Размер шрифта */
        color: #800040; /* Цвет текста */
        margin-top: 0; /* Отступ сверху */
    }
    #container {
        width: 500px; /* Ширина слоя */
        margin: 0 auto; /* Выравнивание по центру */
        background: #f0f0f0; /* Цвет фона левой колонки */
    }
    #header {
        font-size: 2.2em; /* Размер текста */
        text-align: center; /* Выравнивание по центру */
        padding: 5px; /* Отступы вокруг текста */
        background: #8fa09b; /* Цвет фона шапки */
        color: #ffe; /* Цвет текста */
    }
    #sidebar {
        margin-top: 10px;
        width: 110px; /* Ширина слоя */
        padding: 0 10px; /* Отступы вокруг текста */
        float: left; /* Обтекание по правому краю */
    }
    #content {
        margin-left: 130px; /* Отступ слева */
        padding: 10px; /* Поля вокруг текста */
        background: #fff; /* Цвет фона правой колонки */
    }
    #footer {
        background: #8fa09b; /* Цвет фона подвала */
        color: #fff; /* Цвет текста */
        padding: 5px; /* Отступы вокруг текста */
        clear: left; /* Отменяем действие float */
    }

```



```

#nav, #aside {
float: left; width: 110px; padding: 5px;
}
#content {
background: #54B948;
margin: 0 120px; /* Ширина колонок */
padding: 10px;
}
#aside {
float: right; color: #000;
}
h2 { margin: 0 0 1em; }
#footer {
clear: both; /* Отменяем действие float */
padding: 5px; /* Отступы вокруг текста */
background: #D71920; /* Цвет фона подвала */
}
</style>
</head>
<body>
<div id="container">
<div id="header">Header</div>
<div id="nav">
<p>
<a href="stat.html">Menu 1</a>
</p>
<p>
<a href="interview.html">Menu 2</a>
</p>
<p>
<a href="ask.html">menu 3</a>
</p>
</div>
<div id="aside">
<h3>Статистика 1</h3>
...
<h3>Статистика 2</h3>
...
<h3>Статистика 3</h3>
...
<h3>Статистика 4</h3>
...
</div>
<div id="content">
<h2>...</h2>
...
<h2>...</h2>
...
<h2>...</h2>
...
<h2>...</h2>
...
<h2>...</h2>
...
</div>
<div id="footer">&copy;...</div>
</div>
</body>
</html>

```


Практическая часть

1. Ознакомится с теоретическим материалом.
2. Используя результат лабораторной работы №1, вы должны переверстать свои страницы в блочном стиле и «раскрасить» их средствами CSS.