

Лабораторная работа №1. Вёрстка персонального сайта.

Результатом работы должны быть 4-5 html страниц. Вёрстка должна быть выполнена по единому шаблону (за исключением главной страницы). Вёрстка должна быть выполнена в семантическом стиле («дизайн» и пр. оформление – запрещено). Графические элементы (изображения) – допускаются.

Вёрстка

Вёрстка в простом понимании это *процесс превращения работы дизайнера в веб-страницу*, отображаемую в браузере. Но один и тот же результат можно получить разными методами и уже от вёрсталщика зависит, какой из них предпочесть. От такого выбора зависит результат работы сайта, быстрота его отображения, доступность для разных устройств и браузеров. *В настоящей лабораторной работе предлагается идти от исторических основ вёрстки – вёрстки на таблицах.*

В настоящее время уже можно заявить, что табличная вёрстка в большинстве своём является архаизмом. Тем не менее, существует ещё множество сайтов, свёрстанных именно при содействии таблиц. Такой консерватизм связан со следующими обстоятельствами.

- Вёрстка слоями сложнее, чем таблицами, поскольку требует от разработчика глубоких знаний спецификации CSS.
- Браузеры содержат разные ошибки при интерпретации стилевых свойств, поэтому следует знать особенности поведения основных браузеров, их ошибки и уметь обходить их.
- Практически каждую задачу можно решить несколькими способами, от разработчика требуется выбрать оптимальный, т.е. наименее затратный по времени и усилиям. Для оценки оптимальности необходим практический опыт, чтобы иметь в запасе разные методы, применяемые в конкретных ситуациях.

Макеты

Макет из двух колонок

Двухколодная модульная сетка достаточно часто применяется на сайтах, при этом, как правило, в одной колонке располагается основной материал (текст статьи, например), а во второй — ссылки на разделы сайта и другая информация. Для создания подобного макета таблицы достаточно удобны — каждая ячейка выступает в качестве отдельной колонки, что позволяет легко регулировать различные параметры отображения документа.

Для начала рассмотрим самый простой вариант, когда ширина левой колонки жёстко задана в пикселах, а ширина правой колонки варьируется в зависимости от размеров окна браузера. Для этого требуется задать общую ширину всей таблицы в процентах через атрибут `width` тега `<table>` и для первой ячейки установить её ширину в пикселах или процентах также с помощью атрибута `width`, но уже для тега `<td>`:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Две колонки</title>
  </head>
  <body>
    <table width="100%" cellpadding="5">
```

```

<tr>
  <td width="200" valign="top">Левая колонка</td>
  <td valign="top">Правая колонка</td>
</tr>
</table>
</body>
</html>

```

Макет из трёх колонок

Использование трёх колонок на страницах сайта обусловлено шириной информации, которую требуется показать посетителю. Обычно одна колонка, самая широкая, отдаётся под текст статьи, а остальные колонки применяются для ссылок, рекламы, анонсов и др.

Принцип создания трехколонной модульной сетки с помощью таблицы аналогичен созданию двухколонной сетки, поэтому остановлюсь лишь на некоторых моментах.

Ширина разных колонок зависит от используемого макета — фиксированного или «резинового». При макете фиксированной ширины общая ширина таблицы задается в пикселах и остаётся постоянной независимо от разрешения монитора и размера окна браузера. При этом ширину отдельных колонок также имеет смысл установить в пикселах. В примере ширина таблицы задана как 950 пикселей, а колонок соответственно 150, 600 и 200 пикселей

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Три колонки</title>
  </head>
  <body>
    <table width="950" cellspacing="0">
      <tr>
        <td width="150">Колонка 1</td>
        <td width="600">Колонка 2</td>
        <td width="200">Колонка 3</td>
      </tr>
    </table>
  </body>
</html>

```

При «резиновом» макете ширина таблицы устанавливается в процентах от ширины окна браузера и, таким образом, напрямую зависит от неё. Здесь возможны два варианта:

- ширина всех ячеек задана в процентах;
- сочетание процентов и пикселей, когда ширина одних ячеек устанавливается в процентах, а других — в пикселах.

В первом случае вначале устанавливается ширина всей таблицы в процентах, а затем ширина отдельных ячеек. Причём в сумме ширина ячеек должна получиться 100%, несмотря на то, что размер таблицы может быть иным. Дело в том, что ширина таблицы вычисляется относительно доступного пространства веб-страницы, а размер ячеек устанавливается относительно всей таблицы в целом.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Три колонки</title>
  </head>

```

```

<body>
  <table width="100%" cellspacing="0">
    <tr>
      <td width="10%">Колонка 1</td>
      <td width="60%">Колонка 2</td>
      <td width="30%">Колонка 3</td>
    </tr>
  </table>
</body>
</html>

```

Рассмотрим два основных варианта, когда для задания ширины колонок одновременно применяются проценты и пиксели. Первый вариант состоит в том, что размер крайних колонок устанавливается в пикселах, а ширина средней колонки вычисляется автоматически, исходя из заданной ширины таблицы, рисунок 1.1:

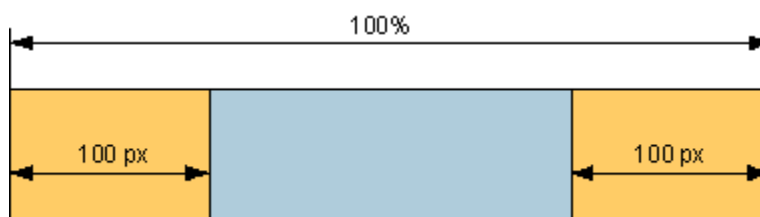


Рис. 1.1 Ширина средней колонки определяется браузером

Для создания подобного макета понадобится таблица с тремя ячейками. Ширину первой и третьей ячейки устанавливаем в пикселах, а ширину средней ячейки намеренно не задаём. При этом обязательно следует определить общую ширину всей таблицы.

Во втором варианте ширина двух колонок устанавливается в процентах, а третьей — в пикселах. В подобном случае обойтись одной таблицей не удастся, рисунок 1.2.

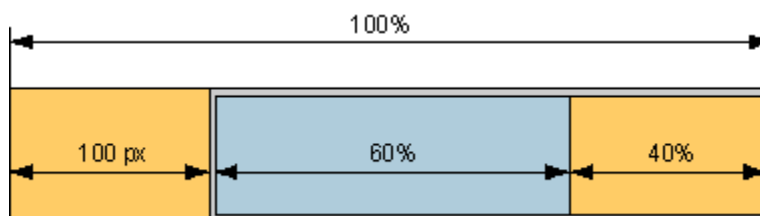


Рис. 1.2. Применение вложенных таблиц

Вначале создаём таблицу заданного размера и с двумя ячейками. Левая ячейка будет выступать в роли первой колонки, и для неё устанавливаем требуемую ширину в пикселах. Ширину для правой ячейки не определяем, поэтому она будет занимать оставшееся пространство, а также служить каркасом для других колонок. Внутри этой ячейки добавляем вторую таблицу, тоже состоящую из двух ячеек. И уже для них определяем ширину в процентах.

Справочник по основным тегам

Блочные теги

Блочные элементы характеризуются тем, что занимают всю доступную ширину, высота элемента определяется его содержимым, и он всегда начинается с новой строки.

<blockquote>

Предназначен для выделения длинных цитат внутри документа. Текст, обозначенный этим тегом, традиционно отображается как выровненный блок с отступами слева и справа (примерно по 40 пикселей), а также с пустым пространством сверху и снизу.

<div>

Тег <div> относится к универсальным блочным контейнерам и применяется в тех случаях, где нужны блочные элементы без дополнительных свойств. Также с помощью тега <div> можно выравнивать текст внутри этого контейнера с помощью атрибута align.

<h1>, ..., <h6>

Эта группа тегов определяет текстовые заголовки разного уровня, которые показывают относительную важность секции, расположенной после заголовка.

<hr>

Рисует горизонтальную линию, которая по своему виду зависит от используемых атрибутов. Линия всегда начинается с новой строки, а после неё все элементы отображаются на следующей строке.

<p>

Определяет параграф (абзац) текста.

Строчные теги

Строчными называются такие элементы веб-страницы, которые являются непосредственной частью другого элемента, например, текстового абзаца. В основном они используются для изменения вида текста или его логического выделения.

<a>

Тег <a> является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов name или href тег <a> устанавливает ссылку или якорь.

Определяет жирное начертание шрифта.

<big>

Тег <big> увеличивает размер шрифта на единицу по сравнению с обычным текстом. В HTML размер шрифта измеряется в условных единицах от 1 до 7, средний размер текста, используемый по умолчанию, принят 3. Таким образом, добавление тега <big> увеличивает текст на одну условную единицу.

**
**

Тег
 устанавливает перевод строки в том месте, где этот тег находится. В отличие от тега параграфа <p>, использование тега
 не добавляет пустой отступ перед строкой.

Тег предназначен для акцентирования текста. Браузеры отображают такой текст курсивным начертанием.

<i>

Устанавливает курсивное начертание шрифта.

Тег предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег в контейнер <a>. При этом вокруг изображения отображается рамка, которую можно убрать, добавив атрибут border="0" в тег .

<small>

Тег <small> уменьшает размер шрифта на единицу по сравнению с обычным текстом. По своему действию похож на тег <big>, но действует с точностью до наоборот.

Универсальный тег, предназначенный для определения строчного элемента внутри документа.

Тег предназначен для акцентирования текста. Браузеры отображают такой текст жирным начертанием.

<sub>

Отображает шрифт в виде нижнего индекса. Текст при этом располагается ниже базовой линии остальных символов строки и уменьшенного размера — H_2O .

<sup>

Отображает шрифт в виде верхнего индекса. По своему действию похож на <sub>, но текст отображается выше базовой линии текста — m^2 .

Разница между блочными и строчными элементами следующая

- Строчные элементы могут содержать только данные или другие строчные элементы, а в блочные допустимо вкладывать другие блочные элементы, строчные элементы, а также данные. Иными словами, строчные элементы никак не могут хранить блочные элементы.
- Блочные элементы всегда начинаются с новой строки, а строчные таким способом не акцентируются.
- Блочные элементы занимают всю доступную ширину, например, окна браузера, а ширина строчных элементов равна их содержимому плюс значения отступов, полей и границ.

Теги для списков

Списком называется взаимосвязанный набор отдельных фраз или предложений, которые начинаются с маркера или цифры. Списки предоставляют возможность упорядочить и систематизировать разные данные и представить их в наглядном и удобном для пользователя виде.

Тег устанавливает нумерованный список, т.е. каждый элемент списка начинается с числа или буквы и увеличивается по нарастающей.

Устанавливает маркированный список, каждый элемент которого начинается с небольшого символа — маркера.

Тег `` определяет отдельный элемент списка. Внешний тег `` или `` устанавливает тип списка — маркированный или нумерованный.

<dd>, <dt>, <dl>

Тройка элементов предназначена для создания списка определений. Каждый такой список начинается с контейнера `<dl>`, куда входит тег `<dt>` создающий термин и тег `<dd>` задающий определение этого термина. Закрывающий тег `</dd>` не обязателен, поскольку следующий тег сообщает о завершении предыдущего элемента. Тем не менее, хорошим стилем является закрывать все теги.

Теги для таблиц

Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. Обычно таблицы используются для упорядочения и представления табличных данных.

<table>

Служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов `<tr>` и `<td>`.

<td>

Предназначен для создания одной ячейки таблицы. Тег `<td>` должен размещаться внутри контейнера `<tr>`, который в свою очередь располагается внутри тега `<table>`.

<th>

Тег `<th>` предназначен для создания одной ячейки таблицы, которая обозначается как заголовочная. Текст в такой ячейке отображается браузером обычно жирным шрифтом и выравнивается по центру.

<tr>

Тег `<tr>` служит контейнером для создания строки таблицы.

Практическая часть

1. Ознакомится с теоретическим материалом.
2. Результатом работы должны быть 4-5 html страниц, рассказывающих о себе (обучающемся).
3. Вёрстка должна быть выполнена по единому шаблону (за исключением главной страницы).
4. Вёрстка должна быть выполнена в семантическом стиле ¹(«дизайн» и пр. оформление – запрещено). Графические элементы (изображения) – допускаются.
5. В настоящей лабораторной работе предлагается идти от исторических основ вёрстки – вёрстки на таблицах.
6. При разработке необходимо в обязательном порядке использовать ряд указанных тегов. Наборы этих тегов указаны в индивидуальном задании. Таблица 1.1.
7. Использовать макет из двух — (1) или из трёх — (2) колонок.
8. Использовать блочные теги:
`<blockquote>`, `<div>`, `<p>` — (1),
`<h1>`,...,`<h6>` (один или несколько из этих тегов) и `<blockquote>` — (2),
`<hr>`, `<blockquote>` — (3).

1 Семантическая вёрстка — подход к разметке, который опирается не на содержание сайта, а на смысловое предназначение каждого блока и логическую структуру документа. Например, заголовки разных уровней — помогают читателю выстроить в голове структуру документа.

9. Использовать строчные теги:

<big>, , ,
, <a>, , , <i> — (1),
<small>, , ,
, <a>, , , <i> — (2),
, , ,
, <a>, , , <i> — (3),
<sub>, , ,
, <a>, , , <i> — (4),
<sup>, , ,
, <a>, , , <i> — (5).

10. Использовать теги для списков:

, — (1),
, — (2),
<dd>, <dt>, <dl> — (3),
<table>, <td>, <th>, <tr> — (4),
<table>, <td>, <tr> — (5).

Задания на лабораторную работу

Таблица 1.1

№	Макет (П.7)	Блочные теги (П.8)	Строчные теги (П.9)	Теги списков (П.10)
1.	2	1	4	2
2.	1	1	5	5
3.	2	2	2	4
4.	1	3	5	2
5.	1	2	4	2
6.	2	1	4	4
7.	2	2	3	3
8.	1	2	3	5
9.	2	2	2	4
10.	1	2	3	4
11.	1	2	4	5
12.	2	3	4	3
13.	2	3	1	4
14.	1	1	1	3
15.	2	3	3	4
16.	1	3	3	4
17.	2	2	3	2
18.	1	3	1	2
19.	2	3	4	5
20.	1	2	2	4
21.	2	2	3	3
22.	1	2	1	2
23.	2	3	3	4
24.	1	1	2	2
25.	1	1	3	3
26.	1	3	5	2
27.	2	1	4	4
28.	1	3	4	1
29.	2	1	4	3
30.	1	3	5	3
31.	1	1	4	2
32.	1	2	3	2
33.	1	2	3	3
34.	2	1	4	3
35.	1	2	4	5
36.	1	2	3	4
37.	2	3	3	2
38.	1	1	4	5
39.	1	2	5	4
40.	2	2	3	5
41.	1	2	5	2
42.	2	1	4	2
43.	2	3	4	5
44.	2	1	1	2
45.	1	1	5	3
46.	1	2	3	3
47.	1	3	2	1
48.	2	3	2	3
49.	1	3	1	3
50.	1	2	4	4

51.	1	1	2	3
52.	2	2	5	2
53.	1	3	5	5
54.	2	3	4	5
55.	2	3	4	5
56.	2	2	3	1
57.	2	1	2	5
58.	2	3	4	1
59.	2	1	4	4
60.	2	2	4	3
61.	1	2	4	3
62.	2	2	4	1
63.	1	3	4	2
64.	1	3	2	4
65.	1	2	2	3
66.	2	1	3	1
67.	1	1	3	4
68.	2	3	1	4
69.	1	2	4	5
70.	1	3	3	2
71.	2	2	2	3
72.	2	1	2	4
73.	2	2	3	5
74.	2	2	2	2
75.	2	2	3	2
76.	2	1	2	3
77.	2	1	2	4
78.	2	1	2	4
79.	1	2	3	3
80.	1	3	3	3
81.	2	2	3	3
82.	1	2	2	2
83.	2	2	5	5
84.	2	3	2	3
85.	2	2	2	4
86.	2	1	4	2
87.	1	2	3	4
88.	2	1	1	4
89.	1	1	3	3
90.	2	3	5	4
91.	1	2	2	2
92.	2	2	4	2
93.	1	2	3	3
94.	1	2	4	1
95.	1	2	2	2
96.	1	2	4	1
97.	1	1	4	3
98.	2	3	5	4
99.	1	3	4	4
100.	2	1	2	2