

CREATING WEB-PAGES

Using HTML5 and CSS3

HTML



CSS



Урок 7

Таблицы

Содержание

Таблицы	3
Создание простейшей таблицы.	3
Объединение ячеек: атрибуты <code>colspan</code> , <code>rowspan</code> ..	14
Тег заголовка: <code><caption></code> . Теги логического группирования столбцов: <code><colgroup></code> , <code><col></code> . Теги логического структурирования таблиц: <code><thead></code> , <code><tbody></code> , <code><tfoot></code>	22
Управление рамками таблицы.	28
Практика: создание сложных таблиц	29
Табличная верстка — что это и почему не используется в современном стандарте?	36
Домашнее задание	40

Материалы урока прикреплены к данному PDF-файлу. Для доступа к материалам, урок необходимо открыть в программе Adobe Acrobat Reader.

Таблицы

Создание простейшей таблицы

Таблицы в HTML5 используются для представления информации в структурированном виде, который представляет из себя двумерную таблицу, состоящую из строк и столбцов ячеек с данными.

Содержимым ячейки может быть любой контент: заголовки, списки, ссылки, изображения, формы и даже другие таблицы.

Для того чтобы создать таблицу нужно использовать парный тег `<table>`, который представляет из себя контейнер, внутри которого должны находиться все элементы таблицы.

Строки таблицы создаются с помощью парного тега `<tr>`, который так же является контейнером для парных тегов ячеек.

Ячейки таблицы делятся на два типа: ячейки заголовков столбца или строки (тег `<th>`), а также ячейки для отображения данных таблицы (тег `<td>`).

Пример кода структуры простой таблицы на три строки и три столбца:

```
<table>
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
    <th>Column 3</th>
  </tr>
  <tr>
```

```

        <td>Lorem ipsum dolor.</td>
        <td>Lorem ipsum dolor.</td>
        <td>Lorem ipsum dolor.</td>
    </tr>
    <tr>
        <td>Lorem.</td>
        <td>Lorem.</td>
        <td>Lorem.</td>
    </tr>
</table>

```

В браузере мы видим следующее отображение:

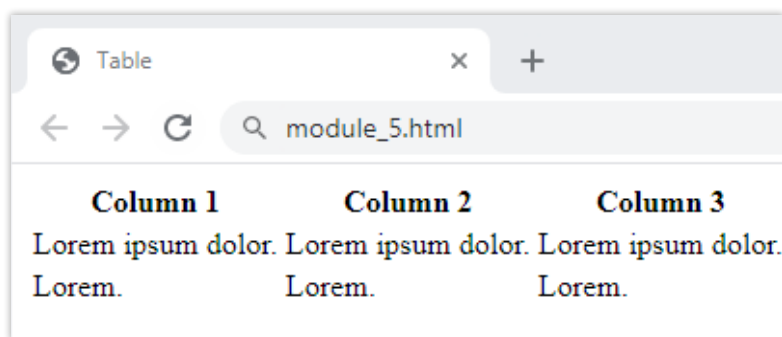


Иллюстрация 1

Ячейки первой строки представлены тегами `<th>`. Содержимое этих ячеек визуально отличается: текст выровнен по середине ячейки и имеет «жирное» начертание.

Содержимое ячеек основного тела таблицы выровнено по левому краю ячейки.

По умолчанию таблицы и ячейки не имеют видимых границ. Для того, чтобы они появились необходимо использовать стилевое свойство `border`, которое используется вместо устаревшего атрибута `border` тега `<table>`:

```

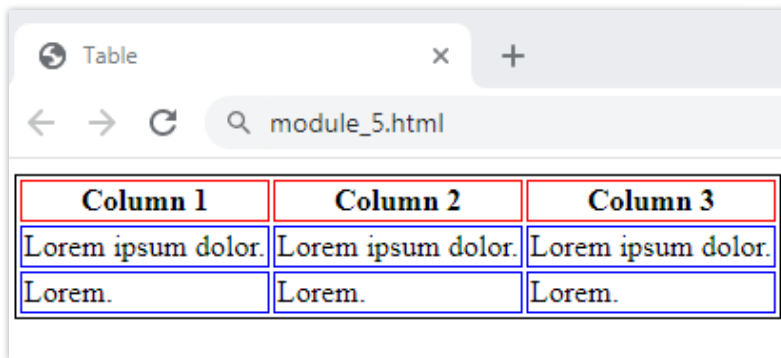
/* внешние границы всей таблицы */
table {border: 1px solid black;}

/* границы ячеек заголовков столбцов */
th {border: 1px solid red;}

/* границы ячеек основной части */
td {border: 1px solid blue;}

```

Теперь таблица имеет следующий вид:



Column 1	Column 2	Column 3
Lorem ipsum dolor.	Lorem ipsum dolor.	Lorem ipsum dolor.
Lorem.	Lorem.	Lorem.

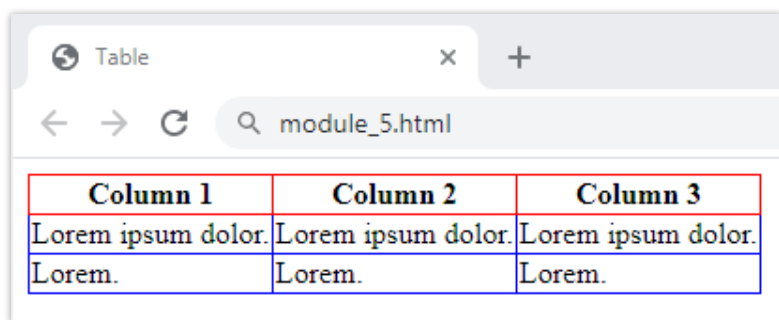
Иллюстрация 2

Между ячейками таблицы имеются промежутки, которые можно убрать с помощью стилевого свойства `border-collapse: collapse;`, которое устанавливается для тега `<table>`.

```

table {
    border: 1px solid black;
    border-collapse: collapse;
}
th {border: 1px solid red;}
td {border: 1px solid blue;}

```



Column 1	Column 2	Column 3
Lorem ipsum dolor.	Lorem ipsum dolor.	Lorem ipsum dolor.
Lorem.	Lorem.	Lorem.

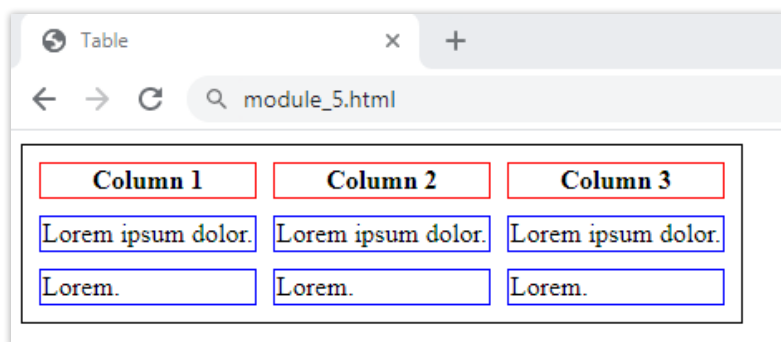
Иллюстрация 3

На рисунке выше видно, что граница самой таблицы при значении `border-collapse: collapse;` не отображается.

Если же необходимо увеличить отступы между ячейками, то используется стилевое свойство `border-spacing`, которое в качестве параметра принимает абсолютные либо относительные единицы измерения. В этом случае свойство `border-collapse` принимает значение по умолчанию `separate`, которое устанавливает отступы вокруг границ ячеек (рис. 4). Например:

```
table {
  border: 1px solid black;
  border-collapse: separate;
  border-spacing: 10px;
}
```

Размер таблицы (ширина и высота) определяется содержимым таблицы. Размер же конкретной ячейки зависит от размера соседних ячеек. Так, ширина любой ячейки будет определяться размером самой широкой ячейки в текущем столбце. Высота любой ячейки будет определяться наибольшей высотой содержимого в текущей строке.



Column 1	Column 2	Column 3
Lorem ipsum dolor.	Lorem ipsum dolor.	Lorem ipsum dolor.
Lorem.	Lorem.	Lorem.

Иллюстрация 4

Для того, чтобы задать размеры всем ячейкам, достаточно задать ширину и высоту одной ячейке в каждой строке и одной ячейке в каждом столбце. Сейчас для этого используют стилевые свойства **width** и **height**, которые заменили устаревшие атрибуты **width** и **height**.

Для удобного обращения к одной ячейке таблицы, пропишем ей идентификатор, который сделает нашу ячейку уникальной.

```
<table>
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
    <th>Column 3</th>
  </tr>
  <tr>
    <td>Lorem ipsum dolor.</td>
    <td id="uniq">Lorem ipsum dolor.</td>
    <td>Lorem ipsum dolor.</td>
  </tr>
  <tr>
    <td>Lorem.</td>
    <td>Lorem.</td>
```

```

        <td>Lorem.</td>
    </tr>
</table>

```

Обратимся к ячейке в стилях и зададим ей ширину и высоту:

```

#uniq{
    width: 30vw;
    height: 20vh;
}

```

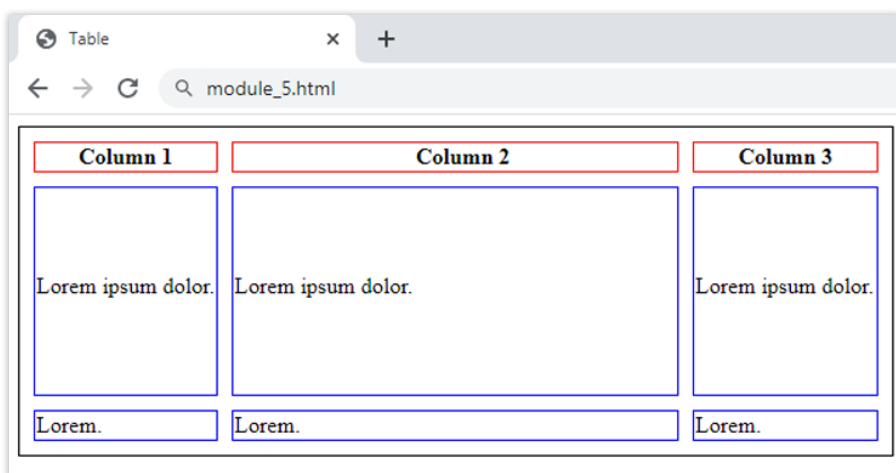


Иллюстрация 5

На рисунке выше видно, что размеры одной ячейки повлияли на размеры всех ячеек, а также на всю таблицу.

Для выравнивания контента внутри ячейки используются стилевые свойства, которые пришли взамен устаревшим атрибутам **align** (горизонтальное выравнивание) и **valign** (вертикальное выравнивание):

- **text-align** используется для выравнивания текста по горизонтали, может принимать значения:
 - ▷ **center** — выравнивает по центру;
 - ▷ **left** — выравнивает по левому краю;
 - ▷ **right** — выравнивает по правому краю;
 - ▷ **justify** — выравнивает по ширине;
 - ▷ **initial** — устанавливает значение по умолчанию;
 - ▷ **inherit** — наследует значение родительского элемента.
- **vertical-align** используется для выравнивания текста, а также строчных элементов внутри ячейки, по вертикали, может принимать значения:
 - ▷ **baseline** — выравнивание по базовой линии (по умолчанию);
 - ▷ **top** — выравнивает по верхнему краю;
 - ▷ **middle** — выравнивает по середине;
 - ▷ **bottom** — выравнивает по нижнему краю;
 - ▷ **initial** — устанавливает значение по умолчанию;
 - ▷ **inherit** — наследует значение родительского элемента.

Выровняем текст нашей уникальной ячейки по правому нижнему краю:

```
#uniq{
  width: 30vw;
  height: 20vh;
  text-align: right;
  vertical-align: bottom;
}
```

Column 1	Column 2	Column 3
Lorem ipsum dolor.	Lorem ipsum dolor. Lorem ipsum dolor.	Lorem ipsum dolor.
Lorem.	Lorem.	Lorem.

Иллюстрация 6

Для того, чтобы изменить фон таблицы, строки или ячейки необходимо в стилевом файле обратиться к нужному элементу и установить ему свойство **background-color**.

```
table {
    border: 1px solid black;
    border-collapse: separate;
    border-spacing: 10px;
    /*фонový цвет для таблицы*/
    background-color: #aaaaaa;
}

tr{
    /*фонový цвет для строки*/
    background-color: #cccccc;
}

th {
    border: 1px solid red;
}
```

```
td {
    border: 1px solid blue;
}

#uniq{
    width: 30vw;
    height: 20vh;
    text-align: right;
    vertical-align: center;
    /*фонový цвет для ячейки*/
    background-color: #eeeeee;
}
```

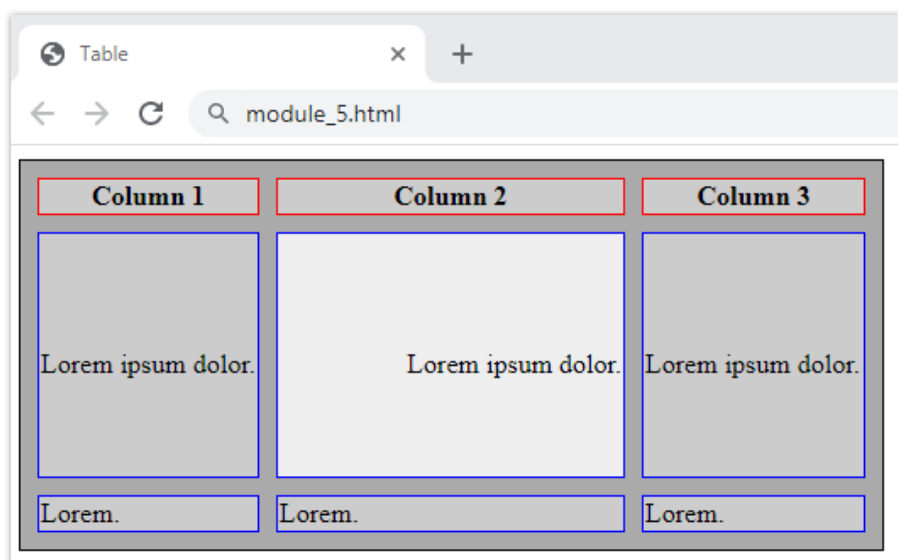


Иллюстрация 7

Для того, чтобы изменить или установить изображение в качестве фона таблицы, строки или ячейки необходимо в стилевом файле обратиться к нужному элементу и установить ему свойство **background-image** (рис. 8).

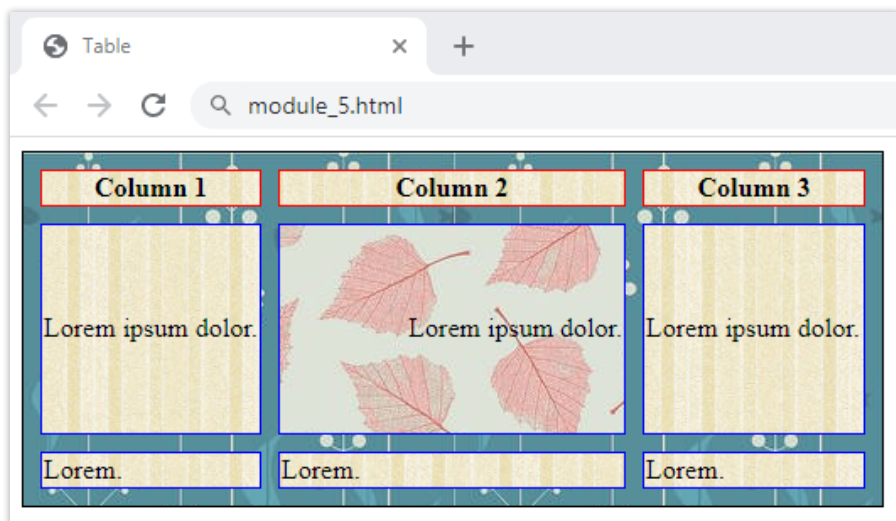


Иллюстрация 8

```
table {
    border: 1px solid black;
    border-collapse: separate;
    border-spacing: 10px;
    /*фоновое изображение для таблицы.
    В случае, если путь к изображению не содержит
    пробелов или специальных символов, можно
    прописывать путь без кавычек*/
    background-image: url(bg_1.jpg);
}

tr{
    /*фоновое изображение для строки*/
    background-image: url(bg_2.jpg);
}

th {
    border: 1px solid red;
}
```

```
td {
    border: 1px solid blue;
}

#uniq{
    width: 30vw;
    height: 20vh;
    text-align: right;
    vertical-align: center;
    /*фоновое изображение для ячейки*/
    background-image: url(bg_3.jpg);
}
```

Если ячейка пустая (без контента), то можно ее скрыть, используя стилевое свойство `empty-cells:hide;`, которое устанавливается для тега `table`.

```
<table>
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
    <th>Column 3</th>
  </tr>

  <tr>
    <td>Lorem ipsum dolor.</td>
    <td id="uniq">Lorem ipsum dolor.</td>
    <td>Lorem ipsum dolor.</td>
  </tr>

  <tr>
    <td></td><!-- пустая ячейка -->
    <td>Lorem.</td>
    <td></td><!-- пустая ячейка -->
  </tr>
</table>
```

```

table {
    border: 1px solid black;
    border-collapse: separate;
    border-spacing: 10px;

    background-image: url(bg_1.jpg);
    /*скрываем пустые ячейки*/
    empty-cells: hide;
}

```

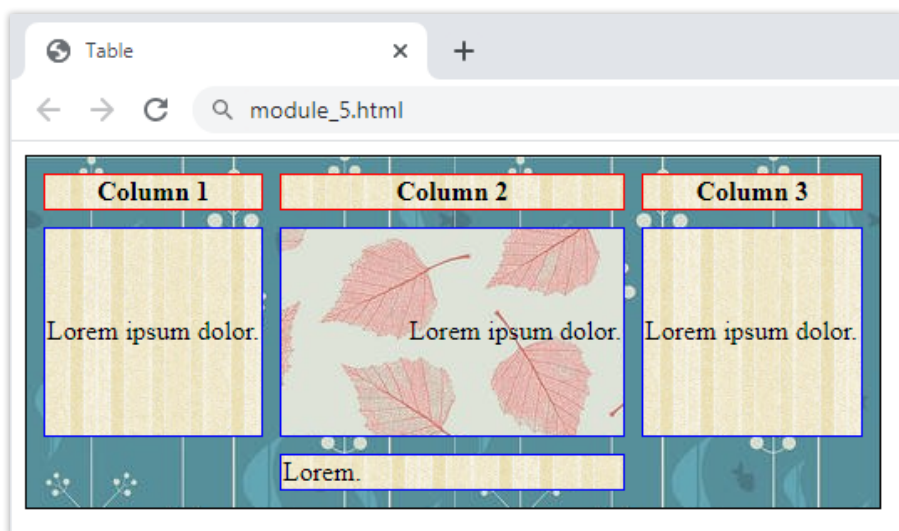


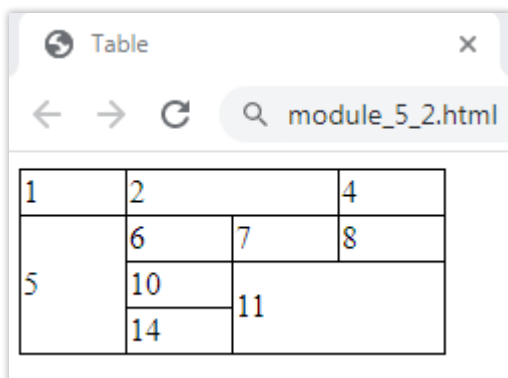
Иллюстрация 9

На рисунке выше видно, что в последней строке не видно ячеек, в которых отсутствует контент.

Объединение ячеек: атрибуты `colspan`, `rowspan`

Иногда ячейки таблицы необходимо объединять, например, в шапке таблицы объединить ячейку для общего наименования столбцов таблицы.

Создадим таблицу, в которой будет 4 строки по 4 ячейки и поэтапно приведем ее к следующему виду:



1	2	3	4
5	6	7	8
5	10	11	11
5	14	11	11

Иллюстрация 10

Ниже представлен начальный код таблицы. Для удобства все ячейки пронумерованы:

```
<table>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
    <td>4</td>
  </tr>
  <tr>
    <td>5</td>
    <td>6</td>
    <td>7</td>
    <td>8</td>
  </tr>
  <tr>
    <td>9</td>
    <td>10</td>
    <td>11</td>
```

```

        <td>12</td>
    </tr>

    <tr>
        <td>13</td>
        <td>14</td>
        <td>15</td>
        <td>16</td>
    </tr>
</table>

```

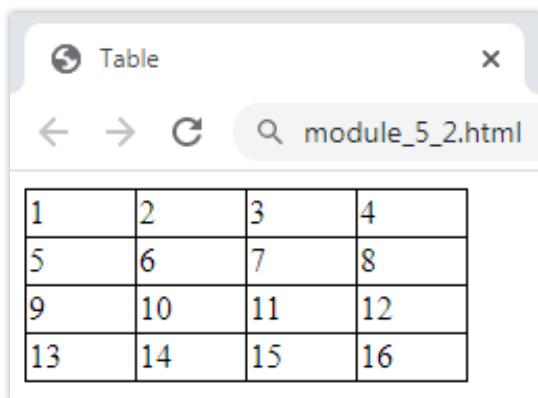
И стилевые свойства:

```

table {
    border-collapse: collapse;
}

td {
    border: 1px solid black;
    width: 50px;
    height: 20px;
}

```



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

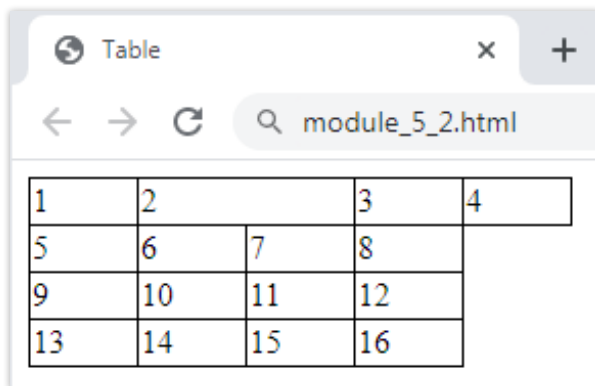
Иллюстрация 11

Для того, чтобы объединить ячейку по горизонтали используется атрибут `colspan`. В качестве значения атрибута `colspan` устанавливается целое число равное количеству объединяемых ячеек по горизонтали в одной строке.

Объединим в первой строке две ячейки (с номером 2 и 3) по горизонтали. Для этого в ячейку с номером 2 пропишем атрибут `colspan="2"`.

```
<tr>
  <td>1</td>
  <td colspan="2">2</td>
  <td>3</td>
  <td>4</td>
</tr>
```

Если мы сейчас посмотрим на таблицу, то увидим, что ячейка с номером 2 объединилась, а ячейка с номером 4 выходит за границы таблицы. Так происходит, потому что эта ячейка стала лишней и заняла место в пятой колонке, которая в нашей таблице отсутствует.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Иллюстрация 12

Для того, чтобы избежать этого, необходимо просто удалить «лишнюю» ячейку или закомментировать. Закомментируем ячейку с номером 3, так как именно ее мы объединили.

```
<tr>
  <td>1</td>
  <td colspan="2">2</td>
  <!-- <td>3</td> -->
  <td>4</td>
</tr>
```

1	2		4
5	6	7	8
9	10	11	12
13	14	15	16

Иллюстрация 13

Для того, чтобы объединить ячейку по вертикали используется атрибут **rowspan**. В качестве значения атрибута **rowspan** устанавливается целое число равное количеству объединяемых ячеек по вертикали в одном столбце.

Объединим три ячейки по вертикали с номерами 5, 9 и 13. Для этого ячейке с номером 5 зададим значение атрибута **rowspan="3"**. Ячейки с номерами 9 и 13 — удалим или закомментируем.

Table

module_5_2.html

1	2		4
5	6	7	8
	10	11	12
	14	15	16

Иллюстрация 14

```

<table>
  <tr>
    <td>1</td>
    <td colspan="2">2</td>
    <!-- <td>3</td> -->
    <td>4</td>
  </tr>

  <tr>
    <td rowspan="3">5</td>
    <td>6</td>
    <td>7</td>
    <td>8</td>
  </tr>

  <tr>
    <!-- <td>9</td> -->
    <td>10</td>
    <td>11</td>
    <td>12</td>
  </tr>

  <tr>
    <!-- <td>13</td> -->

```

```

        <td>14</td>
        <td>15</td>
        <td>16</td>
    </tr>

</table>

```

Можно объединять ячейки одновременно и по вертикали, и по горизонтали, прописывая одной ячейке два атрибута **colspan** и **rowspan**, причем порядок указания атрибутов не важен.

Объединим ячейки с номерами 11, 12, 15 и 16. Для этого ячейке с номером 11 пропишем атрибуты **colspan="2"** и **rowspan="2"**. А остальные ячейки с номерами 12, 15 и 16 удалим или закомментируем.

```

<table>
  <tr>
    <td>1</td>
    <td colspan="2">2</td>
    <!-- <td>3</td> -->
    <td>4</td>
  </tr>

  <tr>
    <td rowspan="3">5</td>
    <td>6</td>
    <td>7</td>
    <td>8</td>
  </tr>

  <tr>
    <!-- <td>9</td> -->
    <td>10</td>

```

```

        <td colspan="2" rowspan="2">11</td>
        <!-- <td>12</td> -->
    </tr>

    <tr>
        <!-- <td>13</td> -->
        <td>14</td>
        <!-- <td>15</td> -->
        <!-- <td>16</td> -->
    </tr>

</table>

```

Вот что у нас получилось в итоге:

1	2		4
5	6	7	8
	10	11	
	14		

Иллюстрация 15

Объединяя ячейки нужно помнить одно основное правило: в каждой строке и в каждом столбце должна быть хотя бы одна необъединенная ячейка. Иначе вы можете столкнуться с ситуацией, когда все ячейки строки станут «лишними» и строка не будет отображаться в таблице.

Тег заголовка: <caption>. Теги логического группирования столбцов: <colgroup>, <col>. Теги логического структурирования таблиц: <thead>, <tbody>, <tfoot>

Для того, чтобы задать заголовок таблице используется тег `<caption>`. Этот тег прописывается сразу же после открывающего тега `<table>` и находится вне строки или ячейки таблицы. Рассмотрим использование тега на примере. Создадим таблицу на 5 строк и 7 столбцов и зададим ей заголовок. Ниже приведен фрагмент кода с заголовком:

```
<table>
  <caption>Заголовок таблицы</caption>
  <tr>
```

В браузере таблица отображается так:

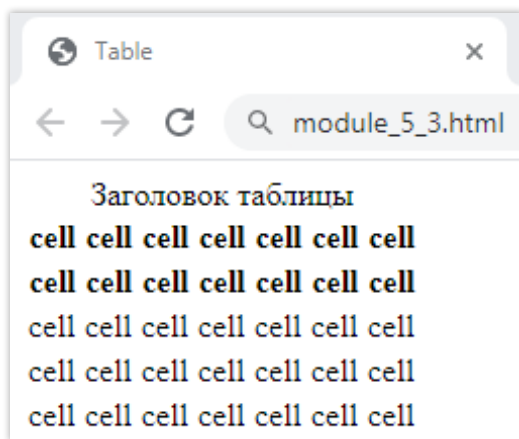


Иллюстрация 16

По умолчанию заголовок таблицы находится сверху таблицы и выровнен по середине таблицы. С помощью

стилей можно изменить положение заголовка. Для выравнивания по горизонтали используется свойство `text-align` со значениями `left` (слева), `center` (по центру) и `right` (справа). Для изменения расположения заголовка над или под таблицей используется свойство `caption-side` со значениями `top` (вверху) и `bottom` (внизу) таблицы. На следующем примере заголовок выровнен по правой стороне и расположен под таблицей:

```
caption{
  text-align: right;
  caption-side: bottom;
}
```

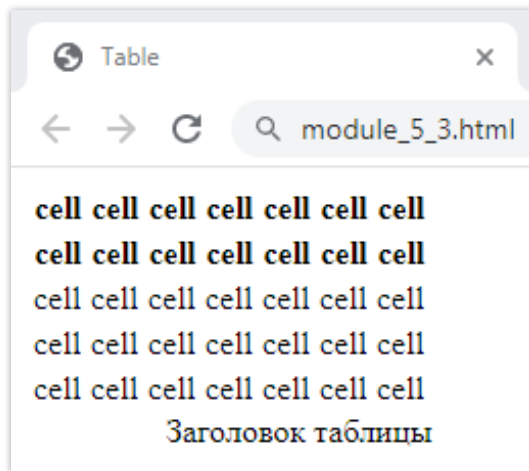


Иллюстрация 17

Часто в таблице необходимо выделять цветом ячейки, содержимое которых имеет общее содержание. Для этой цели существуют теги, которые объединяют ячейки в логические группы.

Для логической группировки столбцов применяются теги `<colgroup>` и `<col>`, которые прописываются внутри тега `<table>`, сразу после тега заголовка `<caption>`, если он имеется. Элементы `<colgroup>` и `<col>` используются для одинакового форматирования ячеек по вертикали.

Тег `<colgroup>` представляет из себя контейнер, в котором содержатся теги `<col>`. Для указания количества объединяемых столбцов применяется атрибут `span`, значением которого является целое число.

Рассмотрим на нашем примере, объединим в три колонки ячейки таблицы следующим образом: в первой колонке будет два столбца, во второй — четыре и в третьей один. Для указания различного фона колонкам, воспользуемся классами.

```
<table>
  <caption>Заголовок таблицы</caption>
  <colgroup>
    <col span="2" class="col-1">
    <col span="4" class="col-2">
    <col class="col-3">
  </colgroup>
</tr>
```

В стилях пропишем фоны для колонок:

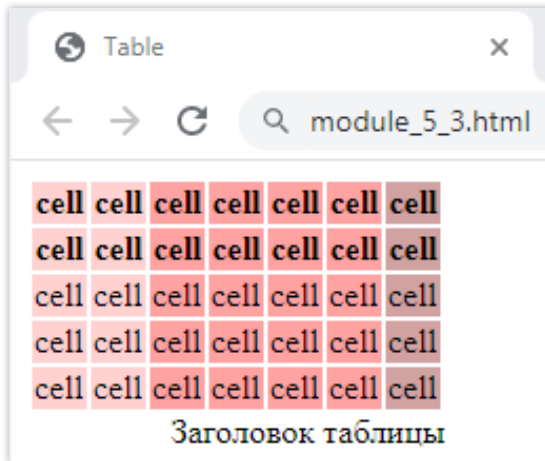
```
.col-1{
  background-color: #ffcccc;
}

.col-2{
  background-color: #ff9999;
}
```



```
.col-3{
    background-color: #cc9999;
}
```

Теперь таблица выглядит так:



cell	cell	cell	cell	cell	cell	cell
cell	cell	cell	cell	cell	cell	cell
cell	cell	cell	cell	cell	cell	cell
cell	cell	cell	cell	cell	cell	cell
cell	cell	cell	cell	cell	cell	cell

Заголовок таблицы

Иллюстрация 18

Для логического структурирования данных таблицы используются теги `<thead>`, `<tbody>`, `<tfoot>`. Рассмотрим каждый из них более подробно.

Тег `<thead>` используется как контейнер для строк таблицы и применяется для группировки строк с ячейками для заголовков столбцов, тем самым формируя «шапку» таблицы. Прописывается внутри тега `<table>` первым либо сразу после тегов `<caption>` и `<colgroup>`, если они имеются. В таблице можно использовать только один элемент `<thead>`.

Тег `<tfoot>` предназначен для формирования «подвала» таблицы, также используется как контейнер для

строк. В таблице можно использовать только один элемент `<tfoot>`, который располагается сразу же после тега `<thead>`.

Тег `<tbody>` используется как контейнер для строк таблицы, которые содержит ячейки с основным контентом таблицы, формируя «тело» таблицы. В таблице можно использовать только один элемент `<tbody>`, который прописывается после тегов `<thead>` и `<tfoot>`.

Рассмотрим использование данных тегов на нашем примере. Полный код таблицы теперь имеет вид:

```
<table>
  <caption>Заголовок таблицы</caption>
  <colgroup>
    <col span="2" class="col-1">
    <col span="4" class="col-2">
    <col class="col-3">
  </colgroup>

  <thead>
    <tr>
      <th>cell</th>
      <th>cell</th>
      <th>cell</th>
      <th>cell</th>
      <th>cell</th>
      <th>cell</th>
      <th>cell</th>
    </tr>
  </thead>

  <tfoot>
    <tr>
      <th>cell</th>
      <th>cell</th>
```

```

        <th>cell</th>
        <th>cell</th>
        <th>cell</th>
        <th>cell</th>
        <th>cell</th>
    </tr>
</tfoot>

<tbody>
    <tr>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
    </tr>
    <tr>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
    </tr>
    <tr>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
        <td>cell</td>
    </tr>
</tbody>
</table>

```

Выделим шапку и подвал, используя стиль для цвета текста:

```
thead{
    color: #ffff00;
}

tfoot{
    color: #ffffff;
}
```

Мы установили желтый цвет текста для «шапки» таблицы и белый цвет текста для «подвала» таблицы. На рисунке ниже видно, что ячейки «подвала» расположены внизу таблицы.

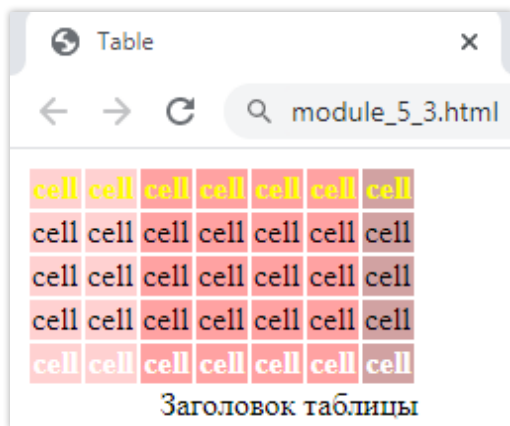


Иллюстрация 19

Управление рамками таблицы

Структурированные данные в виде таблиц использовались с самых истоков возникновения языка html, поэтому у таблицы есть много устаревших атрибутов, среди

которых атрибуты, устанавливающие отображение рамок таблицы. Такими атрибутами являются:

- **border** — для задания толщины границы в пикселях;
- **bordercolor** — для задания цвета границы таблицы;
- **frame** — указывает как отображать границы вокруг таблицы;
- **rules** — устанавливает где отображать границы между ячейками.

Вместо перечисленных атрибутов, необходимо использовать стилевое свойство **border** для установки границы со всех сторон либо стилевые свойства **border-top**, **border-bottom**, **border-left**, **border-right** для указания границы с определенной стороны.

Практика: создание сложных таблиц

В качестве примера создадим сложную таблицу с использованием всех изученных тегов.

TITLE TABLE				
SPECIFICATION	FREE	BASIC	STANDART	PREMIUM
Lorem ipsum dolor	+	+	+	+
Ratione deserunt modi	+	+	+	+
Vel maiores odio	-	+	+	+
Minus vel aperiam natus	-	-	+	+
Provid minima sapiente	-	-	-	+
Eveniet eaque excepturi	-	-	-	+
PRICE	\$ 0.00	\$ 9.99	\$ 19.99	\$ 29.99

Иллюстрация 20

Таблица будет состоять из 8 строк и 5 столбцов. Каждый столбец будет представлять отдельную вертикальную колонку. Также у таблицы будет шапка, подвал и основное тело таблицы. По итогу наша таблица должна будет выглядеть как на рисунке ниже:

Создадим *index.html* и пропишем в нем следующий код:

```
<table>
  <caption>Title table</caption>
  <colgroup>
    <col class="col-1">
    <col class="col-2">
    <col class="col-3">
    <col class="col-4">
    <col class="col-5">
  </colgroup>

  <thead>
    <tr>
      <th>specification</th>
      <th>free</th>
      <th>basic</th>
      <th>standard</th>
      <th>premium</th>
    </tr>
  </thead>

  <tfoot>
    <tr>
      <th>Total</th>
      <th>$ 0.00</th>
      <th>$ 9.99</th>
      <th>$ 19.99</th>
      <th>$ 29.99</th>
    </tr>
  </tfoot>
```

```

<tbody>
  <tr>
    <td>Lorem ipsum dolor</td>
    <td>+</td>
    <td>+</td>
    <td>+</td>
    <td>+</td>
  </tr>

  <tr>
    <td>Ratione deserunt modi</td>
    <td>+</td>
    <td>+</td>
    <td>+</td>
    <td>+</td>
  </tr>

  <tr>
    <td>Vel maiores odio</td>
    <td>-</td>
    <td>+</td>
    <td>+</td>
    <td>+</td>
  </tr>

  <tr>
    <td>Minus vel aperiam natus</td>
    <td>-</td>
    <td>-</td>
    <td>+</td>
    <td>+</td>
  </tr>

  <tr>
    <td>Provid minima sapiente</td>
    <td>-</td>
    <td>-</td>

```

```

        <td>-</td>
        <td>+</td>
    </tr>

    <tr>
        <td>Eveniet eaque excepturi</td>
        <td>-</td>
        <td>-</td>
        <td>-</td>
        <td>+</td>
    </tr>
</tbody>

</table>

```

В браузере таблица выглядит так:

Title table				
specification	free	basic	standart	premium
Lorem ipsum dolor	+	+	+	+
Ratione deserunt modi	+	+	+	+
Vel maiores odio	-	+	+	+
Minus vel aperiam natus	-	-	+	+
Provid minima sapiente	-	-	-	+
Eveniet eaque excepturi	-	-	-	+
Total	\$ 0.00	\$ 9.99	\$ 19.99	\$ 29.99

Иллюстрация 21

Приступим к стилизовому оформлению:

```
table{
    /*устанавливаем ширину таблицы в % от ширины body*/
    width: 80%;
    /*
        чтобы таблица сильно не сжималась и не растягивалась
        при изменении ширины окна браузера, прописываем
        минимальную и максимальную ширину в пикселях
    */
    min-width: 650px;
    max-width: 800px;
    /*располагаем таблицу по середине по горизонтали*/
    margin: 0 auto;
    /*убираем отступы между ячейками*/
    border-collapse: collapse;
    /*устанавливаем системный шрифт для текста*/
    font-family: Arial, Helvetica, sans-serif;
}

caption{
    /*выравниваем заголовок по горизонтали по правому краю*/
    text-align: right;
    /*переводим текст в верхний регистр*/
    text-transform: uppercase;
    /*устанавливаем цвет заголовка таблицы*/
    color: #666666;
    /*задаем жирность шрифту*/
    font-weight: 700;
}

/*
    текст ячеек заголовков столбцов в шапке и подвале
    переводим в верхний регист и задаем белый цвет
*/
th{
    text-transform: uppercase;
    color: #ffffff;
}
```

```

/*задаем оттенок серого цвета первой ячейке заголовков
столбцов в шапке и подвале*/
tr th:first-child{
    color: #666666;
}

/*
    всем ячейкам таблицы устанавливаем выравнивание
    по центру по горизонтали и внутренний отступ сверху
    и снизу
*/
td, th{
    text-align: center;
    padding: 1em 0;
}

/*
    первой ячейке каждой строки задаем выравнивание
    по левой стороне, серый цвет и внутренний отступ
    слева
*/
tr td:first-child{
    text-align: left;
    color: #666666;
    padding-left: 0.7em;
}

/*
    содержимое основных ячеек - элементы span -
    оформляем следующим образом:
*/
span{
    color: #ffffff;
    font-weight: 700;
    font-size: 1.2em;
    display: inline-block;
    border: 2px solid #ffffff;
    border-radius: 50%;

```

```
width: 25px;
height: 25px;
line-height: 25px;
}

/*
каждой колонке, на которые разбили таблицу, задаем
фоновый цвет и ширину, причем четыре колонки будут
иметь одинаковую ширину, а ширина первой колонки будет
высчитываться автоматически
*/
.col-1{
background-color: #dddddd;
width: auto;
}
.col-2{
background-color: #17d0d3;
width: 17%;
}

.col-3{
background-color: #8ac149;
width: 17%;
}

.col-4{
background-color: #feca28;
width: 17%;
}

.col-5{
background-color: #f44236;
width: 17%;
}

thead{
/*задаем нижнюю границу для шапки таблицы*/
border-bottom: 1px solid #ebebeb;
}
```

```
tfoot{
  /*задаем верхнюю границу для подвала таблицы*/
  border-top: 1px solid #ebebeb;
}
```

В итоге таблица приобрела следующий вид:

SPECIFICATION	FREE	BASIC	STANDART	PREMIUM
Lorem ipsum dolor	+	+	+	+
Ratione deserunt modi	+	+	+	+
Vel maiores odio	-	+	+	+
Minus vel aperiam natus	-	-	+	+
Provid minima sapiente	-	-	-	+
Eveniet eaque excepturi	-	-	-	+
PRICE	\$ 0.00	\$ 9.99	\$ 19.99	\$ 29.99

Иллюстрация 22

Табличная верстка — что это и почему не используется в современном стандарте?

Табличная верстка — это способ верстки веб-страницы, в котором в качестве структурной основы используется таблица. При этом тело страницы `<body>` фактически имеет один дочерний элемент `<table>`, ячейки которого определяют «шапку», «подвал» и основную части веб-страницы.

Рассмотрим небольшой пример с использованием табличной верстки. Создадим html страницу со следующим содержимым:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
  <title>Table</title>
  <style>
    table{
      width: 100%;
    }
    th,td{
      border: 1px solid #333;
    }
  </style>
</head>

<body>
  <table>
    <thead>
      <tr>
        <th colspan="3">header</th>
      </tr>
    </thead>
    <tfoot>
      <tr>
        <th colspan="3">footer</th>
      </tr>
    </tfoot>
    <tbody>
      <tr>
        <td width="20%">left</td>
```

```

        <td width="60%">main</td>
        <td width="20%">right</td>
    </tr>
</tbody>
</table>

</body>
</html>

```

В результате веб-страница выглядит как на рисунке ниже. Основные структурные элементы страницы представляют из себя ячейки с соответствующим контентом.

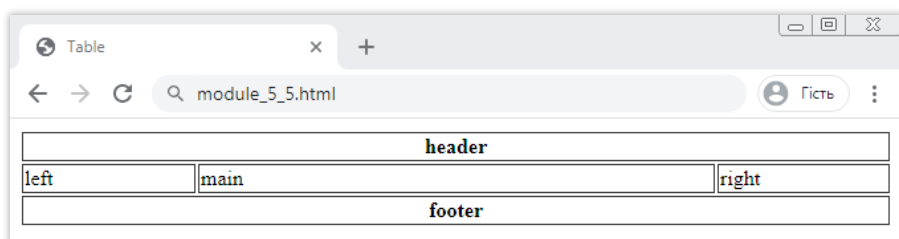


Иллюстрация 23

Табличная верстка широко применялась в то время, когда еще не был определен стандарт CSS, и использование таблицы это был единственный простой способ расположить элементы на странице в определенном порядке. Но в этом случае нарушался основной принцип языка гипертекстовой разметки, где визуальное отображение не должно зависеть от логических элементов структуры страницы. То есть, в случае с табличной версткой, тег `<table>` использовался не по назначению.

С появлением CSS стало возможным разделение логической структуры документа и его визуального

отображения, и появилась блочная верстка, как альтернатива табличной.

Современный стандарт языка HTML5 предполагает использование таблиц только для отображения табличных данных: сравнительных характеристик, различных расписаний, статистик и так далее.

Домашнее задание

Выполнить верстку таблицы:

ASPECT		SIMPLE	PROGRESSIVE	PERFECT	PERFECT PROGRESSIVE
MEANING		<i>a common aspect</i>	<i>a process</i>	<i>priority</i>	<i>priority + process</i>
		When?	At what time?	By what time?	Since what time? How long?
Period of time		<i>usually, often, always, seldom, every day (week, month, year)</i>	<i>now, at the moment</i>	<i>ever, never, just, already, not...yet, by 3 p.m.</i>	<i>since 3 p.m., for a long time, for a month</i>
Present	+	V/Vs	am/is/are + Ving	have/has + Ved/V ₃	have/has + been + Ving
	?	do/does + V	inversion	inversion	inversion
	-	do/does + not + V	am/is/are + not + Ving	have/has + not + Ved/V ₃	have/has + not + been + Ving
Period of time		<i>yesterday, last week (month, year), long ago</i>	<i>yesterday at 3p.m., yesterday from 6 till 7, when you came</i>	<i>yesterday by 3p.m., before some time in the past</i>	<i>yesterday since 3p.m., for some time in the past</i>
Past	+	Ved/V ₂	was/were + Ving	had + Ved/V ₃	had + been + Ving
	?	did + V	inversion	inversion	inversion
	-	did + not + V	was/were + not + Ving	had + not + Ved/V ₃	had + not + been + Ving
Period of time		<i>tomorrow, next week (month, year)</i>	<i>tomorrow at 3p.m., tomorrow from 6 till 7, when you come</i>	<i>tomorrow by 3p.m., before some time in the future</i>	<i>tomorrow since 3p.m., for some time in the future</i>
Future	+	will + V	will + be + Ving	will + have + Ved/V ₃	will + have + been + Ving
	?	inversion	inversion	inversion	inversion
	-	won't + V	won't + be + Ving	won't + have + Ved/V ₃	won't + have + been + Ving

Иллюстрация 24



Урок 7

Таблицы

© Здановская Елена.

© STEP IT Academy, www.itstep.org

Все права на охраняемые авторским правом фото-, аудио- и видеопроизведения, фрагменты которых использованы в материале, принадлежат их законным владельцам. Фрагменты произведений используются в иллюстративных целях в объёме, оправданном поставленной задачей, в рамках учебного процесса и в учебных целях, в соответствии со ст. 1274 ч. 4 ГК РФ и ст. 21 и 23 Закона Украины «Про авторське право і суміжні права». Объём и способ цитируемых произведений соответствует принятым нормам, не наносит ущерба нормальному использованию объектов авторского права и не ущемляет законные интересы автора и правообладателей. Цитируемые фрагменты произведений на момент использования не могут быть заменены альтернативными, не охраняемыми авторским правом аналогами, и как таковые соответствуют критериям добросовестного использования и честного использования.

Все права защищены. Полное или частичное копирование материалов запрещено. Согласование использования произведений или их фрагментов производится с авторами и правообладателями. Согласованное использование материалов возможно только при указании источника.

Ответственность за несанкционированное копирование и коммерческое использование материалов определяется действующим законодательством Украины.