**Организация данных с помощью таблиц**

[Главная](https://webref.ru/) / [Вёрстка](https://webref.ru/layout) / [Уроки по HTML и CSS](https://webref.ru/layout/learn-html-css) / Организация данных с помощью таблиц

Таблицы в HTML были созданы, чтобы обеспечить простой способ разметки структурированных табличных данных и для отображения этих данных в форме, которую пользователь легко читает и воспринимает.

Когда HTML был ещё в процессе разработки, CSS, однако, не поддерживался широко в браузерах, так что таблицы были основным средством, с помощью которого создавались сайты. Они применялись для позиционирования содержимого, а также для построения общего макета страницы. Это работало в то время, но это не то, для чего таблицы предназначались, что и привело ко множеству связанных проблем.

К счастью, мы прошли с тех пор долгий путь. Сегодня таблицы используются специально для организации данных (как и должно быть), а CSS свободно работает над позиционированием и компоновкой.

Построение табличных данных по-прежнему является испытанием. Как таблица должна быть построена в HTML во многом зависит от данных и как они будет отображаться. Затем, когда они размечены в HTML, таблицы должны быть оформлены через CSS, чтобы сделать информацию более чёткой и понятной для пользователей.

**Создание таблицы**

Таблицы состоят из данных, содержащихся в столбцах и строках и HTML предлагает несколько разных элементов для их определения и структурирования. Как минимум таблица должна состоять из элементов <table>, <tr> (table row, строка таблицы) и <td> (table data, данные таблицы). Для улучшения структуры и дополнительного семантического значения таблицы могут включать в себя элемент <th> (table header, заголовок таблицы), а также несколько других элементов. Когда все эти элементы работают вместе, они образуют цельную таблицу.

**<table>**

Мы применяем элемент <table> для инициализации таблицы на странице. Использование данного элемента означает, что информация внутри этого элемента будет отображаться как табличная в виде необходимых столбцов и строк.

<table>

...

</table>

**Строки таблицы**

После того, как таблица была определена в HTML, строки таблицы могут быть добавлены с помощью элемента <tr>. Таблица может содержать множество строк или элементов <tr>. В зависимости от количества информации для отображения, количество строк таблицы может быть значительным.

<table>

<tr>

...

</tr>

<tr>

...

</tr>

</table>

**Данные таблицы**

После того, как таблица определена и были созданы строки, в таблицу могут быть добавлены ячейки данных с помощью элемента <td>. Перечисление нескольких элементов <td> друг за другом создаст столбцы в строке таблицы.

<table>

<tr>

<td>Не заставляйте меня думать Стив Круг</td>

<td>На складе</td>

<td>1</td>

<td>$30.02</td>

</tr>

<tr>

<td>UX-дизайн. Практическое руководство по проектированию опыта взаимодействия Расс Унгер, Кэролайн Чендлер</td>

<td>На складе</td>

<td>2</td>

<td>$52.94 ($26.47 &#215; 2)</td>

</tr>

<tr>

<td>Изучаем HTML5 Брюс Лоусон, Реми Шарп</td>

<td>Нет на складе</td>

<td>1</td>

<td>$22.23</td>

</tr>

<tr>

<td>Пуленепробиваемый веб-дизайн Дэн Сидерхолм</td>

<td>На складе</td>

<td>1</td>

<td>$30.17</td>

</tr>

</table>

**Демонстрация таблицы**

**Заголовок таблицы**

Чтобы назначить заголовок для столбца или строки ячеек, применяется элемент <th>. Он работает как элемент <td> в том смысле, что создаёт ячейку данных. Отличие элемента <th> от <td> заключается в том, что элемент заголовка таблицы содержит смысловое значение, означающее что данные в ячейке являются заголовком, в то время как элемент <td> представляет собой лишь общий фрагмент данных.

Разница между этими двумя элементами похожа на разницу между заголовками (элементы от <h1> до <h6>) и абзацами (элемент <р>). Хотя содержимое заголовка можно поместить внутрь абзаца, не имеет смысла делать это. Потому что применение заголовка добавляет больше семантического смысла содержимому. То же самое верно и для заголовков таблицы.

Заголовки таблицы могут быть задействованы в столбцах и строках; данные в таблице определяют, где заголовки уместны. Атрибут scope помогает точно определить, какое содержимое связано с заголовком. Атрибут scope со значениями col, row, colgroup и rowgroup у элемента <th> применяется к строке или столбцу. Наиболее используемые значения col и row. Значение col указывает, что каждая ячейка внутри столбца имеет непосредственное отношение к этому заголовку таблицы, а значение row указывает, что каждая ячейка в строке относится непосредственно к этому заголовку таблицы.

Использование элемента <th>, наряду с атрибутом scope, чрезвычайно помогает экранным ридерам и вспомогательным технологиям понимать смысл таблицы. Таким образом, они также полезны для тех, кто просматривает веб-страницы с помощью этих технологий.

Кроме того, в зависимости от браузера, заголовки таблицы могут получить некоторый стиль по умолчанию, как правило, жирный и по центру.

<table>

<tr>

<th scope="col">Название</th>

<th scope="col">Наличие</th>

<th scope="col">Кол-во</th>

<th scope="col">Цена</th>

</tr>

<tr>

<td>Не заставляйте меня думать Стив Круг</td>

<td>На складе</td>

<td>1</td>

<td>$30.02</td>

</tr>

<tr>

<td>UX-дизайн. Практическое руководство по проектированию опыта взаимодействия Расс Унгер, Кэролайн Чендлер</td>

<td>На складе</td>

<td>2</td>

<td>$52.94 ($26.47 &#215; 2)</td>

</tr>

<tr>

<td>Изучаем HTML5 Брюс Лоусон, Реми Шарп</td>

<td>Нет на складе</td>

<td>1</td>

<td>$22.23</td>

</tr>

<tr>

<td>Пуленепробиваемый веб-дизайн Дэн Сидерхолм</td>

<td>На складе</td>

<td>1</td>

<td>$30.17</td>

</tr>

</table>

**Демонстрация заголовка таблицы**

Превращение данных в таблицу — это лишь начало. Пока мы только немного прикоснулись к тому, как семантически добавить данные в таблицу, мы можем ещё много что сделать для определения структуры наших таблиц.

**Атрибут headers**

Атрибут headers очень похож на атрибут scope. По умолчанию, атрибут scope может быть использован только для элемента <th>. В случае ячеек, вроде элементов <td> или <th>, которые надо связать с другим заголовком, в игру вступает атрибут headers. Значение атрибута headers для элемента <td> или <th> должно соответствовать значению атрибута id в <th>, который связан с ячейкой.

**Структура таблицы**

Знания о построении таблицы и систематизации данных крайне мощные, однако есть несколько дополнительных элементов, которые помогут нам организовать данные и структуру таблицы. Эти элементы включают <caption>, <thead>, <tbody> и <tfoot>.

**<caption>**

Элемент <caption> обеспечивает подпись или название таблицы. Подпись поможет пользователям определить, что относится к таблице и какие ожидаемые данные они могут в ней найти. <caption> должен следовать сразу же после открывающего тега <table> и располагается в верхней части таблицы по умолчанию.

<table>

<caption>Книги по дизайну и фронтенд-разработке</caption>

...

</table>

**Демонстрация названия таблицы**

**<thead>, <tbody> и <tfoot>**

Содержимое таблиц может быть разбито на несколько групп, включая шапку, тело и подвал. Элементы <thead> (шапка таблицы), <tbody> (тело таблицы) и <tfoot> (подвал таблицы) помогают структурно организовать таблицы.

Шапка таблицы, <thead>, обёртывает строку заголовка или строки таблицы, чтобы обозначить их шапкой. <thead> должен быть помещён в верхней части таблицы, после любого элемента <caption> и перед любым элементом <tbody>.

После <thead> могут идти элементы <tbody> или <tfoot>. Изначально, элемент <tfoot> должен идти сразу после элемента <thead>, но HTML5 предоставил в этом свободу действий. Эти элементы могут теперь встречаться в любом порядке, но они никогда не должны быть родительскими элементами друг другу. Элемент <tbody> должен содержать основные данные таблицы, в то время как <tfoot> содержит описание данных таблицы.

<table>

<caption>Книги по дизайну и фронтенд-разработке</caption>

<thead>

<tr>

<th scope="col">Название</th>

<th scope="col">Наличие</th>

<th scope="col">Кол-во</th>

<th scope="col">Цена</th>

</tr>

</thead>

<tbody>

<tr>

<td>Не заставляйте меня думать Стив Круг</td>

<td>На складе</td>

<td>1</td>

<td>$30.02</td>

</tr>

...

</tbody>

<tfoot>

<tr>

<td>Сумма</td>

<td></td>

<td></td>

<td>$135.36</td>

</tr>

<tr>

<td>НДС</td>

<td></td>

<td></td>

<td>$13.54</td>

</tr>

<tr>

<td>Итого</td>

<td></td>

<td></td>

<td>$148.90</td>

</tr>

</tfoot>

</table>

**Демонстрация группирования элементов таблицы**

**Объединение нескольких ячеек**

Часто две или более ячейки должны быть объединены в одну, не нарушая при этом общей компоновки строк и столбцов. Возможно две ячейки рядом друг с другом содержат одинаковые данные, есть пустая ячейка или ячейки должны быть объединены в целях стилизации. В таких случаях мы можем использовать атрибуты colspan и rowspan. Эти два атрибута работают с элементами <td> или <th>.

Атрибут colspan применяется для получения одной ячейки из нескольких столбцов в таблице, в то время как атрибут rowspan используется, чтобы получить одну ячейку из нескольких строк. Каждый атрибут принимает целое значение, которое указывает количество ячеек для охвата, где 1 — значение по умолчанию.

Используя прежнюю таблицу с книгами, теперь мы можем удалить пустые ячейки в подвале.

<table>

<caption>Книги по дизайну и фронтенд-разработке</caption>

<thead>

<tr>

<th scope="col" colspan="2">Название</th>

<th scope="col">Кол-во</th>

<th scope="col">Цена</th>

</tr>

</thead>

<tbody>

<tr>

<td>Не заставляйте меня думать Стив Круг</td>

<td>На складе</td>

<td>1</td>

<td>$30.02</td>

</tr>

...

</tbody>

<tfoot>

<tr>

<td colspan="3">Сумма</td>

<td>$135.36</td>

</tr>

<tr>

<td colspan="3">НДС</td>

<td>$13.54</td>

</tr>

<tr>

<td colspan="3">Итого</td>

<td>$148.90</td>

</tr>

</tfoot>

</table>

**Демонстрация объединения ячеек**

**Границы в таблице**

Эффективное использование границ может помочь сделать таблицу более наглядной. Границы вокруг таблицы или отдельных ячеек могут оказать большое влияние, когда пользователь пытается интерпретировать данные и быстро сканировать информацию. При стилизации границ через CSS есть два свойства, которые могут быстро пригодиться: border-collapse и border-spacing.

**Свойство border-collapse**

Таблицы состоят из родительского элемента <table>, а также вложенных элементов <th> или <td>. Когда мы устанавливаем границы к этим элементам, то границы вокруг одного элемента начинают соединяться с границами другого элемента. Например, если положить двухпиксельную границу вокруг всей таблицы, а затем дополнительно двухпиксельную границу вокруг каждой ячейки таблицы, то получится четырёхпиксельная граница вокруг каждой ячейки в таблице.

Свойство border-collapse определяет модель границы в таблице. Есть три значения для свойства border-collapse: collapse, separate и inherit. Значение по умолчанию separate у свойства border-collapse означает, что все разные границы складываются друг с другом, как описано выше. Значение collapse, с другой стороны, сжимает границы в одну, выбирая ячейку таблицы в качестве основной.

table {

border-collapse: collapse;

}

th,

td {

border: 1px solid #cecfd5;

padding: 10px 15px;

}

**Демонстрация border-collapse**

**Свойство border-spacing**

Когда свойство border-collapse со значением separate позволяет соединять одну границу с другой, так свойство border-spacing задаёт, какое расстояние, если оно есть, отображается между этими границами.

Например, таблица с однопиксельной границей вокруг всей таблицы и однопиксельной границей вокруг каждой ячейки будет иметь двухпиксельную границу вокруг каждой ячейки, потому что границы складываются друг с другом. Добавление border-spacing со значением 4px отделяет границы на 4 пикселя.

table {

border-collapse: separate;

border-spacing: 4px;

}

table,

th,

td {

border: 1px solid #cecfd5;

}

th,

td {

padding: 10px 15px;

}

**Демонстрация border-spacing**

Свойство border-spacing работает только тогда, когда значение свойства border-collapse задано как separate, это значение по умолчанию. Если свойство border-collapse ранее не указывалось, мы можем использовать свойство border-spacing не волнуясь.

Кроме того, свойство border-spacing может принимать два значения размера: первое значение для горизонтального расстояния, а второе — для вертикального. К примеру, запись border-spacing: 5px 10px установит 5 пикселей горизонтального расстояния между границами и 10 пикселей — вертикального.

**Добавление границ к строкам**

Добавление границ к таблице временами может быть сложным, особенно при вставке границ между строк. В таблице границы не могут применяться к элементам <tr> или структурным элементам таблицы, так что когда мы желаем поставить границу между строк, потребуется некоторое размышление.

Начнём с того, что установим свойство border-collapse для таблицы как collapse, а затем добавим нижний border к каждой ячейке таблицы, независимо от того, какой это элемент, <th> или <td>. По желанию мы можем убрать нижнюю границу у ячеек в последней строке таблицы с помощью псевдокласса :last-child, чтобы выбрать последний элемент <tr> в таблице и нацелиться на элементы <td> внутри строки. Кроме того, если таблица использует структурные элементы, мы должны убедиться, что последняя строка таблицы находится в элементе <tfoot>.

table {

border-collapse: collapse;

}

th,

td {

border-bottom: 1px solid #cecfd5;

padding: 10px 15px;

}

tfoot tr:last-child td {

border-bottom: 0;

}

**Демонстрация добавления границ к строкам таблицы**

**Чередование в таблице**

Среди усилий сделать таблицы более наглядными, одной типовой практикой дизайна является «чередование» строк таблицы с переменным цветом фона. Это делает строки чётче и обеспечивает наглядность при сканировании информации. Один из способов сделать это — поместить класс к каждому элементу <tr> через один и установить цвет фона для этого класса. Другой, более простой способ заключается в использовании псевдокласса :nth-child с параметром even или odd, чтобы выбрать каждый элемент <tr> через один.

Здесь наша таблица с книгами использует псевдокласс :nth-child с параметром even для выбора всех чётных строк таблицы и применения к ним серого фона. Следовательно, каждая строка через одну в теле таблицы будет серой.

table {

border-collapse: separate;

border-spacing: 0;

}

th,

td {

padding: 10px 15px;

}

thead {

background: #395870;

color: #fff;

}

tbody tr:nth-child(even) {

background: #f0f0f2;

}

td {

border-bottom: 1px solid #cecfd5;

border-right: 1px solid #cecfd5;

}

td:first-child {

border-left: 1px solid #cecfd5;

}

**Демонстрация чередования строк в таблице**

В этом коде есть несколько нюансов, о которых стоит упомянуть. Для начала, элемент <table> явно содержит свойство border-collapse со значением separate, а border-spacing задано как 0. Причина в том, что элементы <td> включают в себя границы, в то время как элементы <th> нет. Без свойства border-collapse как separate границы у элементов <td> сделают тело и подвал таблицы шире, чем шапка.

Поскольку свойство border-collapse задано как separate, мы должны быть осторожны в том, как границы применяются к элементам <td>. Здесь границы устанавливаются справа и снизу у всех элементов <td>. Затем первый элемент <td> в <tr> получает левую границу. Так как все элементы <td> соединяются вместе с их границами, это обеспечивает появление сплошной границы вокруг каждого элемента.

Наконец, все элементы <th> получают синий фон, а каждый чётный элемент <tr> получает серый background через псевдокласс :nth-child.

**Выравнивание текста**

В дополнение к границам и чередованию, важную роль в формировании таблицы играет выравнивание текста внутри ячеек. Имена, описания и тому подобное, как правило, выравниваются по левому краю, в то время как номера и другие числа выравниваются по правому. Другая информация, в зависимости от контекста, может быть по центру. Мы можем переместить текст по горизонтали с помощью свойства text-align в CSS, как мы рассмотрели это в уроке 6, «[Работа с типографикой](https://webref.ru/layout/learn-html-css/working-with-typography)».

Для выравнивания текста по вертикали, однако, применяется свойство vertical-align. Это свойство работает только со строчными элементами и ячейками таблицы и не будет работать для блочных, строчно-блочных или ещё каких-либо других элементов.

Свойство vertical-align принимает несколько разных значений, самые популярные — top, middle и bottom. Эти значения вертикально позиционируют текст относительно ячейки таблицы или ближайшего родительского элемента для строчных элементов.

Исправив HTML и CSS и включив свойства text-align и vertical-align, мы можем очистить макет нашей таблицы с книгами. Обратите внимание, что данные в таблице становится намного яснее и удобнее.

HTML

<table>

<thead>

<tr>

<th scope="col" colspan="2">Название</th>

<th scope="col">Кол-во</th>

<th scope="col">Цена</th>

</tr>

</thead>

<tbody>

<tr>

<td>

<strong class="book-title">Не заставляйте меня думать</strong> Стив Круг

</td>

<td class="item-stock">На складе</td>

<td class="item-qty">1</td>

<td class="item-price">$30.02</td>

</tr>

<tr>

<td>

<strong class="book-title">UX-дизайн. Практическое руководство по проектированию опыта взаимодействия</strong> Расс Унгер, Кэролайн Чендлер

</td>

<td class="item-stock">На складе</td>

<td class="item-qty">2</td>

<td class="item-price">$52.94 <span class="item-multiple">$26.47 &#215; 2</span></td>

</tr>

<tr>

<td>

<strong class="book-title">Изучаем HTML5</strong> Брюс Лоусон, Реми Шарп

</td>

<td class="item-stock">Нет на складе</td>

<td class="item-qty">1</td>

<td class="item-price">$22.23</td>

</tr>

<tr>

<td>

<strong class="book-title">Пуленепробиваемый веб-дизайн</strong> Дэн Сидерхолм

</td>

<td class="item-stock">На складе</td>

<td class="item-qty">1</td>

<td class="item-price">$30.17</td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="3">Сумма</td>

<td>$135.36</td>

</tr>

<tr>

<td colspan="3">НДС</td>

<td>$13.54</td>

</tr>

<tr>

<td colspan="3">Итого</td>

<td>$148.90</td>

</tr>

</tfoot>

</table>

CSS

table {

border-collapse: separate;

border-spacing: 0;

color: #4a4a4d;

font: 14px/1.4 "Helvetica Neue", Helvetica, Arial, sans-serif;

}

th,

td {

padding: 10px 15px;

vertical-align: middle;

}

thead {

background: #395870;

color: #fff;

}

th:first-child {

text-align: left;

}

tbody tr:nth-child(even) {

background: #f0f0f2;

}

td {

border-bottom: 1px solid #cecfd5;

border-right: 1px solid #cecfd5;

}

td:first-child {

border-left: 1px solid #cecfd5;

}

.book-title {

color: #395870;

display: block;

}

.item-stock,

.item-qty {

text-align: center;

}

.item-price {

text-align: right;

}

.item-multiple {

display: block;

}

tfoot {

text-align: right;

}

tfoot tr:last-child {

background: #f0f0f2;

}

**Демонстрация выравнивания текста в таблице**

**Полностью стилизованная таблица**

Пока наша таблица с книгами выглядит довольно хорошо. Давайте сделаем ещё один шаг вперёд, скруглим некоторые уголки и чуть больше стилизуем часть текста.

HTML

<table>

<thead>

<tr>

<th scope="col" colspan="2">Название</th>

<th scope="col">Кол-во</th>

<th scope="col">Цена</th>

</tr>

</thead>

<tbody>

<tr>

<td>

<strong class="book-title">Не заставляйте меня думать</strong>

<span class="text-offset">Стив Круг</span>

</td>

<td class="item-stock">На складе</td>

<td class="item-qty">1</td>

<td class="item-price">$30.02</td>

</tr>

<tr>

<td>

<strong class="book-title">UX-дизайн. Практическое руководство по проектированию опыта взаимодействия</strong>

<span class="text-offset">Расс Унгер, Кэролайн Чендлер</span>

</td>

<td class="item-stock">На складе</td>

<td class="item-qty">2</td>

<td class="item-price">$52.94 <span class="text-offset item-multiple">$26.47 &#215; 2</span></td>

</tr>

<tr>

<td>

<strong class="book-title"> Изучаем HTML5</strong>

<span class="text-offset">Брюс Лоусон, Реми Шарп</span>

</td>

<td class="item-stock">Нет на складе</td>

<td class="item-qty">1</td>

<td class="item-price">$22.23</td>

</tr>

<tr>

<td>

<strong class="book-title">Пуленепробиваемый веб-дизайн</strong>

<span class="text-offset">Дэн Сидерхолм</span>

</td>

<td class="item-stock">На складе</td>

<td class="item-qty">1</td>

<td class="item-price">$30.17</td>

</tr>

</tbody>

<tfoot>

<tr class="text-offset">

<td colspan="3">Сумма</td>

<td>$135.36</td>

</tr>

<tr class="text-offset">

<td colspan="3">НДС</td>

<td>$13.54</td>

</tr>

<tr>

<td colspan="3">Итого</td>

<td>$148.90</td>

</tr>

</tfoot>

</table>

CSS

table {

border-collapse: separate;

border-spacing: 0;

color: #4a4a4d;

font: 14px/1.4 "Helvetica Neue", Helvetica, Arial, sans-serif;

}

th,

td {

padding: 10px 15px;

vertical-align: middle;

}

thead {

background: #395870;

background: linear-gradient(#49708f, #293f50);

color: #fff;

font-size: 11px;

text-transform: uppercase;

}

th:first-child {

border-top-left-radius: 5px;

text-align: left;

}

th:last-child {

border-top-right-radius: 5px;

}

tbody tr:nth-child(even) {

background: #f0f0f2;

}

td {

border-bottom: 1px solid #cecfd5;

border-right: 1px solid #cecfd5;

}

td:first-child {

border-left: 1px solid #cecfd5;

}

.book-title {

color: #395870;

display: block;

}

.text-offset {

color: #7c7c80;

font-size: 12px;

}

.item-stock,

.item-qty {

text-align: center;

}

.item-price {

text-align: right;

}

.item-multiple {

display: block;

}

tfoot {

text-align: right;

}

tfoot tr:last-child {

background: #f0f0f2;

color: #395870;

font-weight: bold;

}

tfoot tr:last-child td:first-child {

border-bottom-left-radius: 5px;

}

tfoot tr:last-child td:last-child {

border-bottom-right-radius: 5px;

}

**Демонстрация стилизации таблицы**

**На практике**

Теперь, когда мы знаем, как создать и стилизовать таблицы, давайте завершим последнюю оставшуюся страницу нашего сайта Styles Conference — расписание.

1. Мы начнём нашу страницу Расписание с открытия файла schedule.html и добавления элемента <section> с классом row, так же, как мы это проделали со всеми другими страницами. Внутри этого нового <section> также добавим элемент <div> с классом container.
2. <section class="row">
3. <div class="container">
4. ...
5. </div>

</section>

С помощью этих элементов и классов мы создали новый раздел страницы с белым background и вертикальным padding и выровняли содержимое страницы по центру. Что изменилось здесь по сравнению с другими страницами — класс container на месте класса grid для элемента <div>. Поскольку мы не будем использовать любой класс col-, то отказываемся от класса grid в пользу класса container.

1. В новом разделе мы добавим три таблицы, по одной для каждого дня конференции. Таблицы будут отображать события каждого дня с помощью трёх столбцов и нескольких строк, содержать шапку таблицы и её тело.

Для начала давайте наметим структуру первой таблицы, включая элементы <table>, <thead> и <tbody>.

<section class="row">

<div class="container">

<table>

<thead>

...

</thead>

<tbody>

...

</tbody>

</table>

</div>

</section>

1. В данный момент, хотя наша первая таблица не содержит каких-либо данных, к ней применяются некоторые стили. В частности, сброс, который мы добавили в уроке 1, установил для таблиц свойство border-collapse со значением collapse, а свойство border-spacing со значением 0. Нам нужны эти стили, поэтому оставим их в покое. Тем не менее, создадим новый раздел в нашем файле main.css, чтобы добавить некоторые дополнительные стили.

В нашем новом разделе стилей специально для страницы Расписание (который появится чуть ниже стилей для страницы Спикеры), установим для элементов <table> свойство width как 100% и нижний margin 44 пикселя.

Затем с помощью псевдокласса :last-child определим последний элемент <table> в разделе и установим для него нижний margin 0 пикселей. Это предотвратит конфликт этой таблицы с нижним padding, принадлежащему элементу <section> с классом row.

Пока новый раздел в нашем файле main.css выглядит следующим образом:

/\*

========================================

Расписание

========================================

\*/

table {

margin-bottom: 44px;

width: 100%;

}

table:last-child {

margin-bottom: 0;

}

1. Теперь добавим данные в нашу таблицу. Начнём с первого дня конференции, с семинара 24 августа.

В элементе <thead> у таблицы добавим элемент <tr>. Первой ячейкой в строке будет элемент <th> отмечая центр дня: «Семинары» для этого конкретного дня. Поскольку элемент <th> является заголовком строки, мы также добавим к нему атрибут scope со значением row.

После элемента <th> идёт элемент <td> с датой «24 августа», в данном случае. Теперь у нас чаще будет три столбца, первый из которых служит заголовком таблицы, он определяет время, а вторые два столбца — это обычные ячейки таблицы, которые указывают спикеров на это время. Так как в этой строке нам не нужно два отдельных спикера, мы хотим добавить атрибут colspan со значением 2 к элементу <td>, заставляя его объединить два столбца.

Наш код для таблицы теперь выглядит следующим образом:

<table>

<thead>

<tr>

<th scope="row">

Семинары

</th>

<td colspan="2">

24 августа

</td>

</tr>

</thead>

<tbody>

...

</tbody>

</table>

1. Внутри элемента <tbody> заполним дневные мероприятия. Начнём с добавления элемента <tr> с элементами <th> и <td> непосредственно в строке.

Ко всем последующим элементам <th> мы добавим атрибут scope со значением row, чтобы семантически определить этот элемент в качестве заголовка строки. Затем в элементе <th> добавим элемент <time>, который показывает время первого мероприятия — 8:30, в данном случае. Мы также включим атрибут datetime для элемента <time> со значением времени в часах, минутах и секундах — 08:30:00.

В элемент <td>, который следует за элементом <th>, включим название мероприятия (поскольку в это время нет никаких спикеров) — «Регистрация», в данном случае. Так как существует только одно мероприятие в это время, мы также включим атрибут colspan со значением 2 в элементе <td>.

В целом, код нашей первой таблицы выглядит следующим образом:

<table>

<thead>

<tr>

<th scope="row">

Семинары

</th>

<td colspan="2">

24 августа

</td>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">

<time datetime="08:30:00">8:30</time>

</th>

<td colspan="2">

Регистрация

</td>

</tr>

</tbody>

</table>

1. Для второй строки внутри элемента <tbody> добавим элемент <tr> чуть ниже нашего предыдущего ряда. Затем добавим элемент <th> с атрибутом scope и значением row и снова добавим элемент <time> с соответствующим временем и атрибутом datetime в этом элементе <th>.

После <th> добавим два элемента <td> для двух спикеров, выступающих в одно время. Непосредственно внутри каждого элемента <td> мы добавим элемент <a>, который ссылается туда, где спикер расположен на странице Спикеры. Помните, мы добавили атрибуты id с именем каждого спикера к родительскому элементу? Используя это значение атрибута id, которому предшествует имя файла speakers.html и знак решётки #, мы можем напрямую связать описание доклада спикера с его биографией на странице Спикеры.

Внутрь элемента <a> мы включим элемент <h4> с именем спикера с последующим названием доклада.

Код для первых двух семинаров выглядит следующим образом:

<table>

<thead>

<tr>

<th scope="row">

Семинары

</th>

<td colspan="2">

24 августа

</td>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">

<time datetime="08:30:00">8:30</time>

</th>

<td colspan="2">

Регистрация

</td>

</tr>

<tr>

<th scope="row">

<time datetime="09:00:00">9:00</time>

</th>

<td>

<a href="speakers.html#adam-connor">

<h4>Адам Коннор</h4>

Свет! Камера! Мотор! Вдохновение дизайна от кинематографа

</a>

</td>

<td>

<a href="speakers.html#jennifer-jones">

<h4>Дженифер Джонс</h4>

Чему дизайнеры могут научиться у родителей

</a>

</td>

</tr>

</tbody>

</table>

1. Отсюда мы можем повторить этот шаблон для каждого вида мероприятия и спикера, чтобы закончить нашу первую таблицу, а затем добавить две таблицы для следующих двух дней конференции.

При этом имейте в виду, что шапка таблицы всегда будет включать в себя заголовок таблицы с событием дня и ячейки таблицы, объединяющие две колонки с датой.

Затем в теле каждой таблицы, каждая строка будет содержать заголовок таблицы с временем. После заголовка таблицы идёт мероприятие, спикер или несколько спикеров. Мероприятия без спикеров будут находиться в одном элементе <td>, который объединяет два столбца. Если представлен только один спикер в определённое время, то спикер будет находиться в одном элементе <td>, который объединяет два столбца и содержать элементы <a> и <h4>.

Если есть два спикера одновременно, то каждый спикер будет находиться внутри своего собственного элемента <td>, как и раньше.

Полный код для всех трёх таблиц можно найти в конце этого упражнения. Для справки, таблица для первого дня, который включает в себя обед и более двух спикеров, выглядит следующим образом:

<table>

<thead>

<tr>

<th scope="row">

Семинары

</th>

<td colspan="2">

24 августа

</td>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">

<time datetime="08:30:00">8:30</time>

</th>

<td colspan="2">

Регистрация

</td>

</tr>

<tr>

<th scope="row">

<time datetime="09:00:00">9:00</time>

</th>

<td>

<a href="speakers.html#adam-connor">

<h4>Адам Коннор</h4>

Свет! Камера! Мотор! Вдохновение дизайна от кинематографа

</a>

</td>

<td>

<a href="speakers.html#jennifer-jones">

<h4>Дженифер Джонс</h4>

Чему дизайнеры могут научиться у родителей

</a>

</td>

</tr>

<tr>

<th scope="row">

<time datetime="12:30:00">12:30</time>

</th>

<td colspan="2">

Обед

</td>

</tr>

<tr>

<th scope="row">

<time datetime="14:00:00">14:00</time>

</th>

<td>

<a href="speakers.html#tessa-harmon">

<h4>Тесса Хармон</h4>

Искусство кодирования: Генерация вязанных шаблонов

</a>

</td>

<td>

<a href="speakers.html#russ-unger">

<h4>Расс Унгер</h4>

От Маппетов к мастерству: Основные принципы UX от Джима Хенсона

</a>

</td>

</tr>

</tbody>

</table>

1. Теперь наши таблицы обрели форму, настало время добавить к ним немного стиля. Начнём с добавления некоторых общих стилей для элементов <th> и <td>. Для них добавим нижний padding 22 пикселя и вертикальное выравнивание top. Для элемента <th> специально добавим правый padding 45 пикселей, выравнивание текста как right и width как 20%. Затем для элементов <td> добавим width как 40%.

Ниже наших существующих стилей таблиц и расписания код должен выглядеть следующим образом:

th,

td {

padding-bottom: 22px;

vertical-align: top;

}

th {

padding-right: 45px;

text-align: right;

width: 20%;

}

td {

width: 40%;

}

1. Далее добавим стиль к шапке таблицы и к элементам в ней. Мы установим line-height как 44 пикселя только для элемента <thead>, color как #648880 и font-size как 24 пикселя для всех элементов <th> вложенных в <thead>. Наши новые стили включают следующее:
2. thead {
3. line-height: 44px;
4. }
5. thead th {
6. color: #648880;
7. font-size: 24px;

}

1. Шапка таблицы выглядит хорошо, так что добавим некоторые стили для тела таблицы. Начнём с элементов <th> вложенных в <tbody>: поменяем их цвет, добавим некоторые шрифтовые и текстовые свойства и небольшой верхний padding.
2. tbody th {
3. color: #a9b2b9;
4. font-size: 14px;
5. font-weight: 400;
6. padding-top: 22px;
7. text-transform: uppercase;

}

1. Мы также добавим некоторые стили к элементам <td> вложенных в <tbody>, начиная с верхней границы и padding. Мы стилизуем элементы <td>, которые объединяются в один столбец, добавив к ним padding 15 пикселей справа, чтобы сформировать левый столбец и 15 пикселей padding слева, чтобы сформировать правый столбец. Это установит в общей сложности 30 пикселей padding между двумя столбцами, сохраняя каждую ячейку одинакового размера. Нам не нужно применять левый или правый padding к элементам <td>, которые объединяют два столбца.

Мы добавим все эти горизонтальные padding с помощью псевдоклассов :first-of-type, :last-of-type и :only-of-type. Эти псевдоклассы работают очень похоже на псевдокласс :last-child, который мы использовали ранее.

Псевдокласс :first-of-type выберет первый элемент этого типа в родительском элементе. В нашем случае, селектор td:first-of-type выберет первый элемент <td> внутри элемента <tr>. Затем псевдокласс :last-of-type выберет последний элемент этого типа в родительском элементе.

Опять же, в нашем случае, селектор td:last-of-type выберет последний элемент <td> внутри элемента <tr>. Наконец, псевдокласс :only-of-type выберет элемент, только если это элемент данного типа в родительском элементе. Здесь селектор td:only-of-type выберет только элемент <td>, если это только <td> внутри элемента <tr>, например, когда <td> объединяет два столбца.

Наши стили немного сложные, но они гибкие в удовлетворении потребностей нашей таблицы. Эти новые стили включают следующее:

tbody td {

border-top: 1px solid #dfe2e5;

padding-top: 21px;

}

tbody td:first-of-type {

padding-right: 15px;

}

tbody td:last-of-type {

padding-left: 15px;

}

tbody td:only-of-type {

padding-left: 0;

padding-right: 0;

}

1. Наше расписание и таблица для его отображения соединяются вместе. Давайте настроим несколько стилей для существующих элементов чтобы очистить макет. Начнём с того, что сделаем все ссылки в таблице немного серыми. Если мы нацелимся только на элементы <a> в таблице, наши заголовки с именем спикера внутри ссылки останутся зелёными, в то время как название доклада будет серым, создавая приятный контраст между ними.

В то время как мы настраиваем стили для докладов, также уберём нижний margin для элементов <h4> в таблице, позволяя имени спикера располагаться ближе к заголовку. Мы можем реализовать эти стили с помощью следующего кода:

table a {

color: #888;

}

table h4 {

margin-bottom: 0;

}

1. Наконец, создадим некоторые визуальный контраст между разными видами деятельности происходящих в течение дня. Все доклады хорошо выглядят после наших последних изменений. Для всех других видов деятельности, таких как регистрация, обед и перерывы (которые в теле таблицы), а также для даты (которая в заголовке таблицы) будем использовать светлый серый.

Мы сделаем это, создав новый класс schedule-offset и назначив ему color со значением #a9b2b9.

.schedule-offset {

color: #a9b2b9;

}

После того, как класс находится на месте, добавим его ко всем элементам <td>, которые объединяют два столбца и включают дату, либо мероприятие — регистрацию, обед или перерыв. Вспоминая нашу таблицу для первого дня с семинарами, HTML будет выглядеть следующим образом:

<table>

<thead>

<tr>

<th scope="row">

Семинары

</th>

<td class="schedule-offset" colspan="2">

24 августа

</td>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">

<time datetime="08:30:00">8:30</time>

</th>

<td class="schedule-offset" colspan="2">

Регистрация

</td>

</tr>

...

<tr>

<th scope="row">

<time datetime="12:30:00">12:30</time>

</th>

<td class="schedule-offset" colspan="2">

Обед

</td>

</tr>

...

</tbody>

</table>

Таблицы, которые на первый взгляд выглядят просто, могут оказаться довольно сложными, как в случае с нашим расписанием для Styles Conference. Хорошей новостью является то, что наше расписание теперь завершено и выглядит превосходно.

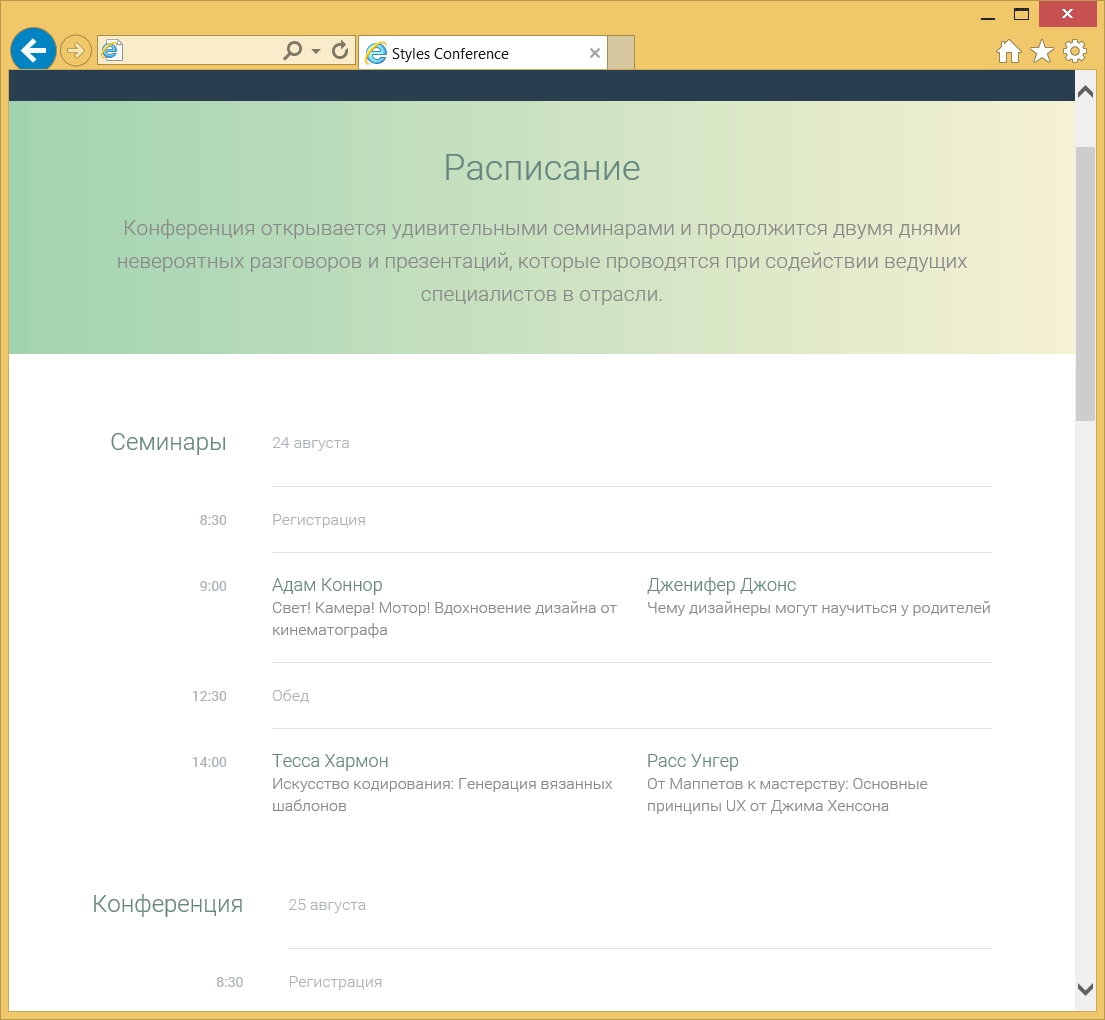


Рис. 11.01. Страница Расписание включает несколько таблиц для Styles Conference

**Демонстрация и исходный код**

Ниже вы можете просмотреть сайт Styles Conference в его нынешнем состоянии, а также скачать исходный код сайта на данный момент.

[Просмотр сайта Styles Conference](https://webref.ru/example/learn-html-css/organizing-data-with-tables/schedule.html) или [Скачать исходный код](https://webref.ru/assets/files/organizing-data-with-tables.zip)

**Резюме**

Ладно, теперь мы знаем, как семантически скомпоновать табличные данные в HTML, также делая это интуитивно с помощью CSS. Обсуждение таблиц было нашим последним основным препятствием при изучении HTML и CSS и мы на этом официально завершили наш сайт Styles Conference.

Для проверки, в этом уроке мы рассмотрели следующее:

* лучшие способы для семантического создания таблиц;
* как сделать, чтобы отдельные ячейки таблицы объединяли несколько столбцов или строк;
* структурные элементы, из которых состоят таблицы;
* разные способы стилизации границ таблицы и как различные свойства влияет на внешний вид таблицы;
* как вертикально выравнивать текст в таблице.

Мы проделали большую работу по применению всех наших новых навыков на практике и ушли далеко вперёд, чем были несколько уроков назад. Давайте закончим на высокой ноте, свяжем некоторые концы и рассмотрим методы написания наилучшего кода.