

Таблицы

Таблицы — страшный сон верстальщика. Они громоздки в своей вёрстке и в них очень легко запутаться. Важно «набить руку» и вы сможете достаточно быстро и без боли верстать даже самые сложные варианты таблиц.

Базовая вёрстка таблиц

Таблица — составной элемент, который формируется сразу из нескольких вложенных друг в друга тегов. Это похоже на то, как верстаются списки — имеется контейнер и внутри него специальные теги-элементы. Таблицы имеют похожие структуры, только контейнеров и элементов немного больше.

Любая таблица начинается с парного тега `<table>`. Одного этого контейнера уже достаточно, чтобы получить первую работающую таблицу.

В отличие от списков, таблицы можно назвать двухмерной системой. Здесь есть строки и столбцы. Именно в таком порядке они и обозначаются.

Для создания строки используется тег `<tr>`. Количество строк и регулируется количеством этих тегов внутри `<table>`. Попробуем создать три строки:

```
<table>
  <tr></tr> <!-- строка 1 -->
  <tr></tr> <!-- строка 2 -->
  <tr></tr> <!-- строка 3 -->
</table>
```

Сейчас эта разметка почти ничего не сообщает браузеру. Если вставить текст в тегах `<tr>`, то браузер посчитает такой код неверным и постарается обработать его самостоятельно. Например, некоторые браузеры автоматически «вытащат» весь этот текст и переместят его за пределы тега `<table>`.

Для добавления данных нужны колонки. Их можно добавить с помощью тега `<td>`. Количество колонок вы так же регулируете самостоятельно. Сколько тегов `<td>`, столько колонок в таблице и будет.

Важно: количество колонок должно совпадать в каждой строке. Если количество колонок будет не совпадать, то таблица может банально «сломаться». Следите за этим.

```
<table>
  <tr>
    <td>Строка 1. Колонка 1.</td>
    <td>Строка 1. Колонка 2.</td>
    <td>Строка 1. Колонка 3.</td>
  </tr>
  <tr>
    <td>Строка 2. Колонка 1.</td>
    <td>Строка 2. Колонка 2.</td>
    <td>Строка 2. Колонка 3.</td>
  </tr>
```

```

<tr>
  <td>Строка 3. Колонка 1.</td>
  <td>Строка 3. Колонка 2.</td>
  <td>Строка 3. Колонка 3.</td>
</tr>
</table>

```

Результат:

HTML	Result	EDIT ON CODEPEN
Строка 1. Колонка 1. Строка 1. Колонка 2. Строка 1. Колонка 3. Строка 2. Колонка 1. Строка 2. Колонка 2. Строка 2. Колонка 3. Строка 3. Колонка 1. Строка 3. Колонка 2. Строка 3. Колонка 3.		

Заметьте, что по умолчанию таблицы не имеют никаких границ между строками и ячейками. Для их добавления можно пойти двумя путями:

1. Указать атрибут `border` для тега `<table>`. Это сразу установит все возможные границы, как для строк, так и для столбцов. Данный метод считается устаревшим и не рекомендуется к использованию.
2. Использовать свойство `border` для необходимых тегов.

Лучший вариант — использование CSS. Так вы будете хранить стили в одном месте, и не будет нужды вспоминать об атрибутах.

```

table,
tr,
td {
  border: 1px solid #000;
}

```

Результат:

HTML

CSS

Result

EDIT ON

CODEPEN

Строка 1. Колонка 1.	Строка 1. Колонка 2.	Строка 1. Колонка 3.
Строка 2. Колонка 1.	Строка 2. Колонка 2.	Строка 3. Колонка 3.
Строка 3. Колонка 1.	Строка 3. Колонка 2.	Строка 3. Колонка 3.

Главная неприятность — возникновение двойных границ. Это может быть дизайнерской «фишкой», но чаще всего такое не требуется. Нужно «схлопнуть» отступы и для этого используется свойство `border-collapse` со значением `collapse`. Для возвращения в исходное состояние используется значение `separate`.

Результат:

HTML

CSS

Result

EDIT ON CODEPEN

LIVE

```
table,
tr,
td {
  border: 1px solid #000;
  border-collapse: collapse;
}

td {
  padding: 10px;
}
```

Строка 1. Колонка 1.	Строка 1. Колонка 2.	Строка 1. Колонка 3.
Строка 2. Колонка 1.	Строка 2. Колонка 2.	Строка 3. Колонка 3.
Строка 3. Колонка 1.	Строка 3. Колонка 2.	Строка 3. Колонка 3.

Выравнивание контента по вертикали

Помимо знакомого вам свойства `text-align`, таблицы позволяют выравнивать контент по вертикали. Это простая операция, которая постоянно использовалась во времена табличной вёрстки. Главное, не пытайтесь найти такие макеты. Выровнять по вертикали можно любой контент, будь то просто текст или какой-то блок.

Для выравнивания по вертикали используется свойство `vertical-align`, которое принимает одно из четырёх значений:

- `baseline` — выравнивание по базовой линии шрифта. Подробнее с таким выравниванием и его принципами вы познакомитесь в курсе [CSS: Flex](#).
- `top` — выравнивание по верхнему краю ячейки.
- `middle` — выравнивание по центру. Стандартное поведение для контента внутри ячеек.
- `bottom` — выравнивание по нижнему краю ячейки.

Результат:

HTML

CSS

Result

EDIT ON CODEPEN

LIVE

```
table {
  border-collapse: collapse;
}

table,
tr,
td {
  border: 1px solid #000;
}

td {
  height: 70px;
  padding: 10px 20px;
}

.vertical-top {
  vertical-align: top;
}
```

Строка 1. Колонка 1.		
	Строка 2. Колонка 2.	
		Строка 3. Колонка 3.

Заголовки таблицы

Сложно представить себе таблицу, которая не имеет заголовков. Без них понять, какая ячейка относится к какой информации почти невозможно. Можно визуально задать стили заголовков и для некоторых ячеек ввести свои стили. Но семантически таких заголовков не будет существовать, что плохо для доступности.

Создать секцию с заголовками можно с помощью тега `<thead>`. Строки и ячейки внутри такой обёртки будут являться заголовками колонок нашей таблицы. Есть и ещё небольшое изменение: вместо тега `<td>` внутри шапки используется тег `<th>`. Это, помимо семантической нагрузки, позволяет проще определять стили.

Хорошей практикой является добавление тега `<tbody>` для основного контента. Если этого не сделать, то браузер подставит его сам, но лучше доверять собственной разметке, чем работе браузера.

В качестве заголовка самой таблицы используется текст, обёрнутый в тег `<caption>`. По умолчанию он располагается сверху всей таблицы, вне зависимости от того, где вы расположите тег. Это поведение можно контролировать с помощью свойства `caption-side`. Оно принимает одно из двух значений:

- `top` — позиция по умолчанию. Заголовок располагается до таблицы.
- `bottom` — расположение заголовка после таблицы.

Пример таблицы с использованием новых тегов и свойств:

html файл

```
<table>
  <caption>Профессии</caption>
  <thead>
    <tr>
      <th>№</th>
      <th>Название профессии</th>
      <th>Описание</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Фронтенд-программист</td>
      <td>JavaScript — язык программирования фронтенд-разработки. С его помощью программисты делают сайты динамичными и «живыми». Самый востребованный и наиболее простой язык для новичков.</td>
    </tr>
  </tbody>
</table>
```

CSS файл

```
html {
  font-family: sans-serif;
}

table {
```

```

border-collapse: collapse;

color: #212529;
}

td,
th {
padding: 0.75rem;
border-top: 1px solid #dee2e6;
}

th {
border-bottom: 2px solid #dee2e6;
}

caption {
margin-top: 25px;

font-size: 0.7em;
text-align: right;

caption-side: bottom;
}

```

Результат:

HTML	CSS	Result	EDIT ON CODEPEN
№	Название профессии	Описание	
1	Фронтенд-программист	JavaScript — язык программирования фронтенд-разработки. С его помощью программисты делают сайты динамичными и «живыми». Самый востребованный и наиболее простой язык для новичков.	Профессии

Объединение строк и столбцов

Не все таблицы имеют простую структуру из ровных столбцов и строк. Объединение нескольких строк или столбцов — распространённая практика при вёрстке таблиц. Рассмотрим базовые правила на примере табеля выплаты зарплаты сотрудников.

Сотрудник	Зарплата	Бонусы	Менеджер
Алексей Примадонин	750\$	63\$	Кодовёнок Хекслетович
Вениамин Редакторович	1200\$	0	
Джедай Падаванов	500\$	100\$	Королева Верстальщина
			Всего: 2613\$

Сама по себе таблица является достаточно простой и не должна вызывать у вас никаких проблем. Попробуйте сверстать её самостоятельно. После этого продолжите урок. Чем больше таблиц вы сверстаете, тем меньше ошибок в них будете допускать.

Создадим базовый каркас всей таблицы, без объединения строк или столбцов:

```
<table>
  <thead>
    <tr>
      <th>Сотрудник</th>
      <th>Зарплата</th>
      <th>Бонусы</th>
      <th>Менеджер</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Алексей Примадонин</td>
      <td>750$</td>
      <td>63$</td>
      <td>Кодовёнок Хекслетович</td>
    </tr>
    <tr>
      <td>Вениамин Редакторович</td>
      <td>1200$</td>
      <td>0</td>
      <td></td>
    </tr>
    <tr>
      <td>Джедай Падаванов</td>
      <td>500$</td>
      <td>100$</td>
      <td>Королева Верстальщина</td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td>Всего: 2613$</td>
    </tr>
  </tbody>
</table>
```

Обратите внимание на пустые ячейки. Они не нужны в данной таблице и от них нужно избавиться таким образом, чтобы информация в соседних ячейках получила правильный перенос и начала занимать всё доступное пространство.

В HTML для этой операции существуют два атрибута:

- **colspan** — объединение столбцов.
- **rowspan** — объединения строк.

Принцип работы достаточно прост — в качестве значения аргумента указывается количество строк/столбцов, которое должен занять элемент. Например,

- **<td colspan="3">** — ячейка должна занять пространство трёх столбцов.
- **<td rowspan="2">** — ячейка должны занять пространство в две строки.

Важно: ячейки, на месте которых будут расположены ячейки с colspan и rowspan, должны быть удалены из таблицы.

Попробуем объединить менеджера первых двух сотрудников. Для этого нужно у ячейки с нужным менеджером установить атрибут **rowspan** со значением 2. Обязательно нужно удалить последнюю ячейку в строке ниже.

```
<tr>
  <td>Алексей Примадонин</td>
  <td>750$</td>
  <td>63$</td>
  <td rowspan="2">Кодовёнок Хекслетович</td>
</tr>
<tr>
  <!-- В этой строке теперь только три столбца -->
  <td>Вениамин Редакторович</td>
  <td>1200$</td>
  <td>0</td>
</tr>
```

Проделаем то же самое для последней строки, только теперь нужно объединить столбцы. Для этого используем атрибут **colspan** со значением 4.

```
<tr>
  <!-- В этой строке всего один столбец, который растянется на 4 -->
  <td colspan="4">Всего: 2613$</td>
</tr>
```

Итоговый вид таблиц:

```
<table>
  <thead>
    <tr>
      <th>Сотрудник</th>
      <th>Зарплата</th>
      <th>Бонусы</th>
      <th>Менеджер</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Алексей Примадонин</td>
      <td>750$</td>
      <td>63$</td>
      <td rowspan="2">Кодовёнок Хекслетович</td>
    </tr>
    <tr>
      <td>Вениамин Редакторович</td>
      <td>1200$</td>
      <td>0</td>
    </tr>
    <tr>
      <td>Джедай Падаванов</td>
      <td>500$</td>
      <td>100$</td>
      <td>Королева Верстальщина</td>
    </tr>
    <tr>
      <td colspan="4" class="text-right text-bold">Всего: 2613$</td>
    </tr>
  </tbody>
</table>
```