Этот раздел познакомит вас с таблицами HTML, представив самые базовые понятия - строки и ячейки, заголовки, слияние строк и столбцов, а также объединение всех ячеек в столбце в целях стилизации.

**Что такое таблица?**

Таблица - это структурированный набор данных, состоящий из строк и столбцов (**табличных данных**). Таблицы позволяют быстро и легко посмотреть значения, показывающие некоторую взаимосвязь между различными типами данных, например - человек и его возраст, или расписание в плавательном бассейне.





Люди постоянно используют таблицы, причем уже давно, как показывает документ по переписи в США, относящийся к 1800 году:



Так что не удивительно, что создатели HTML включили в него средства для структурирования и представления табличных данных в сети.

Как работает таблица?

Смысл таблицы в том, что она жесткая. Информацию легко интерпретировать, визуально сопоставляя заголовки строк и столбцов. Например, посмотрите на приведенную ниже таблицу и найдите единственное (Singular) личное местоимение, используемое в третьем лице ("3rd Person"), с полом (gender) ♀, выступающее в качестве объекта (object) в предложении. Ответ можно найти, сопоставив соответствующие заголовки столбцов и строк.

|  |
| --- |
| **––** |
| **––** | **Subject** | **Object** |
| **Singular** | **1st Person** | I | me |
| **2nd Person** | you | you |
| **3rd Person** | **♂** | he | him |
| **♀** | she | her |
| **o** | it | it |
| **Plural** | **1st Person** | we | us |
| **2nd Person** | you | you |
| **3rd Person** | they | them |
|  |  |  |  |

Если правильно представить таблицу HTML, интерпретировать ее данные смогут даже люди, имеющие проблемы со зрением.

Оформление таблиц

Не питайте ложных иллюзий - чтобы эффективно представлять таблицы в веб, необходимо придать им хорошую структуру в HTML и применить к ним таблицы стилей ([CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS)). В данном разделе мы сфокусируемся на HTML

В этом разделе мы не фокусируемся на CSS, но все же дали простейшую таблицу стилей CSS, чтобы сделать таблицы более читабельными.

Когда не надо использовать таблицы HTML?

HTML-таблицы следует использовать для табличных данных — это то, для чего они предназначены. К сожалению, многие используют таблицы HTML для оформления веб-страниц, например, одна строка для заголовка, одна для содержимого, одна для сносок, и тому подобное. Подробнее об этом можно узнать в [Page Layouts](https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML#Page_layouts) на Accessibility Learning Module. Это происходило из-за плохой поддержки CSS в разных браузерах; в наше время такое встречается гораздо реже, но иногда все же попадается.

Короче говоря, использование таблиц в целях оформления вместо [методов CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout) является плохой идеей по следующим причинам :

1. **Таблицы, используемые для оформления, уменьшают доступность страниц для людей, имеющих проблемы со зрением**: [Скринридеры (Screenreaders](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Cross_browser_testing/Accessibility%22%20%5Cl%20%22Screenreaders)), используемые ими, интерпретируют HTML-теги и читают содержимое пользователю. Поскольку таблицы не являются средством для представления структуры таблицы, и разметка получается сложнее, чем при использовании методов CSS, скринридеры вводят пользователей в заблуждение.
2. **Таблицы создают путаницу тегов**: Как уже упоминалось, оформление страниц с помощью таблиц дает более сложную структуру разметки, чем специально предназначенные для этого методы. Соответственно, такой код труднее писать, поддерживать и отлаживать.
3. **Таблицы не реагируют автоматически на тип устройства**: У надлежащих контейнеров (например, [<header>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/header), [<section>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/section), [<article>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/article), или [<div>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/div)) ширина по умолчанию равна 100% от их родительского элемента. У таблиц же размер по умолчанию подстраивается под их содержимое, так что чтобы они одинаково хорошо работали на разных типах устройств необходими принимать дополнительные меры.

**Упражение: Ваша первая таблица**

Итак, мы уже достаточно говорили о теории, теперь возьмем конкретный пример и построим таблицу.

1. Прежде всего, создайте локальную копию [blank-template.html](https://github.com/mdn/learning-area/blob/master/html/tables/basic/blank-template.html) и [minimal-table.css](https://github.com/mdn/learning-area/blob/master/html/tables/basic/minimal-table.css) в новой папке на вашем компьютере.
2. Содержимое любой таблицы заключается между двумя тегами : [**<table></table>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table). Добавьте их в тело HTML.
3. Самым маленьким контейнером в таблице является ячейка, она создается элементом [**<td>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/td) ('td' - сокращение от 'table data'). Введите внутри тегов table следующее:

<td>Hi, I'm your first cell.</td>

1. Чтобы получить строку из четырех ячеек, необходимо скопировать эти теги три раза. Обновите содержимое таблицы так, чтобы она выглядела следующим образом:
2. <td>Hi, I'm your first cell.</td>
3. <td>I'm your second cell.</td>
4. <td>I'm your third cell.</td>

<td>I'm your fourth cell.</td>

Как видите, ячейки не располагаются одна под другой, на самом деле они автоматически выравниваются по отношению к другим ячейкам той же строки. Каждый элемент <td> создает отдельную ячейку, а все вместе они создают первую строку. Каждая добавленная ячейка удлиняет эту строку.

Чтобы эта строка перестала расти, а новые ячейки перешли на вторую строку, необходимо использовать элемент [**<tr>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/tr) ('tr' - сокращение от 'table row'). Попробуем, как это получится.

1. Поместите четыре уже созданных ячейки между тегами <tr> как здесь показано:
2. <tr>
3. <td>Hi, I'm your first cell.</td>
4. <td>I'm your second cell.</td>
5. <td>I'm your third cell.</td>
6. <td>I'm your fourth cell.</td>

</tr>

1. Теперь, когда одна строка уже есть, добавим еще — каждую строку надо вложить в дополнительный элемент <tr>, а каждая ячейка должна быть внутри элемента <td>.

В результате получится таблица, которая будет выглядеть примерно так:

|  |  |  |  |
| --- | --- | --- | --- |
| Hi, I'm your first cell. | I'm your second cell. | I'm your third cell. | I'm your fourth cell. |
| Second row, first cell. | Cell 2. | Cell 3. | Cell 4. |

**Добавление заголовков с помощью элементов <th>**

Теперь обратимся к табличным заголовкам — особым ячейкам, которые идут вначале строки или столбца и определяют тип данных, которые содержит данная строка или столбец (как "Person" и "Age" в первом примере данной статьи). Чтобы показать, для чего они нужны, возьмем следующий пример. Сначала исходный код:

<table>

 <tr>

 <td>&nbsp;</td>

 <td>Knocky</td>

 <td>Flor</td>

 <td>Ella</td>

 <td>Juan</td>

 </tr>

 <tr>

 <td>Breed</td>

 <td>Jack Russell</td>

 <td>Poodle</td>

 <td>Streetdog</td>

 <td>Cocker Spaniel</td>

 </tr>

 <tr>

 <td>Age</td>

 <td>16</td>

 <td>9</td>

 <td>10</td>

 <td>5</td>

 </tr>

 <tr>

 <td>Owner</td>

 <td>Mother-in-law</td>

 <td>Me</td>

 <td>Me</td>

 <td>Sister-in-law</td>

 </tr>

 <tr>

 <td>Eating Habits</td>

 <td>Eats everyone's leftovers</td>

 <td>Nibbles at food</td>

 <td>Hearty eater</td>

 <td>Will eat till he explodes</td>

 </tr>

</table>

Теперь как выглядит таблица:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Knocky | Flor | Ella | Juan |
| Breed | Jack Russell | Poodle | Streetdog | Cocker Spaniel |
| Age | 16 | 9 | 10 | 5 |
| Owner | Mother-in-law | Me | Me | Sister-in-law |
| Eating Habits | Eats everyone's leftovers | Nibbles at food | Hearty eater | Will eat till he explodes |

Проблема в том, что, хотя вы и можете представить, о чем идет речь, ссылаться на эти данные на так легко, как хотелось бы. Лучше, чтобы строка и столбец с заголовками как-то выделялись.

Упражнение: заголовки

Попробуем улучшить эту таблицу.

1. Сначала создайте локальную копию [dogs-table.html](https://github.com/mdn/learning-area/blob/master/html/tables/basic/dogs-table.html) и [minimal-table.css](https://github.com/mdn/learning-area/blob/master/html/tables/basic/minimal-table.css) в новой папке на вашем компьютере. HTML содержит пример Dogs, который вы уже видели выше.
2. Чтобы опознавать заголовки таблицы в качестве заголовков, визуально и семантически, можно использовать элемент [**<th>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/th) ('th' сокращение от 'table header'). Он работает в точности как <td>, за исключением того, что обозначает заголовок, а не обычную ячейку. Замените в своем HTML все элементы <td>, содержащие заголовки, на элементы <th>.
3. Сохраните HTML и загрузите его в браузер, и вы увидите, что заголовки теперь выглядят как заголовки.

Для чего нужны заголовки?

Мы уже частично ответили на этот вопрос — когда заголовки выделяются, легче искать данные и таблица выглядит лучше.

**Примечание**: По умолчанию к заголовкам таблицы примеряется определенный стиль — они выделены жирным шрифтом и выровнены по центру, даже если вы не задавали для них стиль специально.

Заголовки дают дополнительное преимущество — вместе с атрибутом scope (который мы будем изучать в следующей статье) они помогают улучшить связь каждого заголовка со всеми данными строки или столбца одновременно, что довольно полезно

**Слияние нескольких строк или столбцов**

Иногда нужно бывает, чтобы ячейки распротранялись на несколько строк или столбцов. Возьмем простой пример, в котором приведены имена животных. Иногда бывает нужно вывести имена людей рядом с именами животных. А иногда это не требуется, и тогда мы хотим, чтобы имя животного занимало всю ширину.

Исходная разметка выглядит так:

<table>

 <tr>

 <th>Animals</th>

 </tr>

 <tr>

 <th>Hippopotamus</th>

 </tr>

 <tr>

 <th>Horse</th>

 <td>Mare</td>

 </tr>

 <tr>

 <td>Stallion</td>

 </tr>

 <tr>

 <th>Crocodile</th>

 </tr>

 <tr>

 <th>Chicken</th>

 <td>Cock</td>

 </tr>

 <tr>

 <td>Rooster</td>

 </tr>

</table>

Но результат не такой, как хотелось бы:

|  |
| --- |
| **Animals** |
| **Hippopotamus** |
| **Horse** | Mare |
| Stallion |  |
| **Crocodile** |  |
| **Chicken** | Cock |
| Rooster |  |

Нужно, чтобы  "Animals", "Hippopotamus" и "Crocodile" распространялись на два столбца, а "Horse" и "Chicken" - на две строки. К счастью, табличные заголовки и ячейки имеют атрибуты colspan и rowspan, которые позволяют это сделать. Оба принимают безразмерное числовое значение, которое равно количеству строк или столбцов, на которые должны распространяться ячейки. Например, colspan="2" распространяет ячейку на два столбца.

Воспользуемся colspan и rowspan чтобы улучшить таблицу.

1. Сначала создайте локальную копию [animals-table.html](https://github.com/mdn/learning-area/blob/master/html/tables/basic/animals-table.html) и [minimal-table.css](https://github.com/mdn/learning-area/blob/master/html/tables/basic/minimal-table.css) в новой папке на вашем компьютере. Код HTML содержит пример с животными, который вы уже видели выше.
2. Затем используйте атрибут colspan чтобы распространить "Animals", "Hippopotamus" и "Crocodile" на два столбца.
3. Наконец, используйте атрибут rowspan чтобы распространить  "Horse" и "Chicken" на две строки.

Сохраните код и откройте его в браузере, чтобы увидеть улучшения.

|  |
| --- |
|  |

**Стилизация столбцов**

И последняя возможность, о которой рассказывается в данной статье. HTML позволяет указать, какой стиль нужно применять к целому столбцу данных сразу — для этого применяют элементы  [**<col>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/col) и [**<colgroup>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/colgroup). Их ввели, поскольку задавать стиль для каждой ячейки в отдельности или использовать сложный селектор вроде :nth-child() было бы слишком утомительно.

Возьмем простой пример:

<table>

 <tr>

 <th>Data 1</th>

 <th style="background-color: yellow">Data 2</th>

 </tr>

 <tr>

 <td>Calcutta</td>

 <td style="background-color: yellow">Orange</td>

 </tr>

 <tr>

 <td>Robots</td>

 <td style="background-color: yellow">Jazz</td>

 </tr>

</table>

Что дает нам:

|  |  |
| --- | --- |
| **Data 1** | **Data 2** |
| Calcutta | Orange |
| Robots | Jazz |

Он не идеален, поскольку нам пришлось повторить информацию о стиле для всех стрех ячеек в столбце (в реальном проэкте, возможно, придется вводить class на всех трех и вводит правило в таблице стилей). Вместо этого, мы можем задать информацию один раз, в элементе <col>. Элемент <col> задается в контейнере <colgroup> сразу же за открывающим тегом <table>. Эффект, который мы видели выше, можно задать так:

<table>

 <colgroup>

 <col>

 <col style="background-color: yellow">

 </colgroup>

 <tr>

 <th>Data 1</th>

 <th>Data 2</th>

 </tr>

 <tr>

 <td>Calcutta</td>

 <td>Orange</td>

 </tr>

 <tr>

 <td>Robots</td>

 <td>Jazz</td>

 </tr>

</table>

Мы определяем два "стилизующих столбца". Мы не применяем стиль к первому столбцу, но пустой элемент <col> ввести необходимо — иначе к первому столбцу не будет применен стиль.

Если бы мы хотели применить информацию о стиле к обоим столбцам, мы могли бы просто ввести один элемент <col> с атрибутом span, таким образом:

<colgroup>

 <col style="background-color: yellow" span="2">

</colgroup>

Подобно colspan и rowspan, span принимает безразмерное числовое значение, указывающее, к какому количеству столбцов нужно применить данный стиль.

Упражнение: colgroup и col

Теперь попробуйте сами.

Ниже приведена таблица уроков по языкам. В пятницу (Friday) новый класс целый день изучает голландский (Dutch),  кроме того, во вторник (Tuesday) и четверг (Thursdays) есть занятия по немецкому (German). Учительница хочет выделить столбцы, соответствующие дням, когда она преподает.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Mon** | **Tues** | **Wed** | **Thurs** | **Fri** | **Sat** | **Sun** |
| **1st period** | English |  |  | German | Dutch |  |  |
| **2nd period** | English | English |  | German | Dutch |  |  |
| **3rd period** |  | German |  | German | Dutch |  |  |
| **4th period** |  | English |  | English | Dutch |  |  |

Заново создайте таблицу, проделав указанные ниже действия.

1. Сначала создайте локальную копию файла [timetable.html](https://github.com/mdn/learning-area/blob/master/html/tables/basic/timetable.html) в новой папке на вашем копьютере. Код HTML содержит таблицу, которую вы уже видели выше, но без информации о стиле.
2. Добавьте элемент <colgroup> вверху таблицы, сразу же под тегом <table>, куда вы сможете вставлять элементы <col>.
3. Первые два столбца надо оставить без стиля..
4. Добавьте цвет фона для третьего столбца. Значением атрибута style будет  background-color:#97DB9A;
5. Задайте ширину для четвертого столбца. Значением атрибута style будет width: 42px;
6. Добавьте цвет фона для пятого столбца. Значением атрибута style будет background-color: #97DB9A;
7. Добавьте другой цвет фона и границу для шестого столбца, чтобы показать, что это особый день и она ведет новый класс. Значениями атрибута style будут: background-color:#DCC48E; border:4px solid #C1437A;
8. Последние два дня выходные; значением атрибута style будет width: 42px;

Посмотрите, что у вас получилось.

## Добавление заголовка к таблице с помощью <caption>

Вы можете добавить заголовок для таблицы установив его в элементе [<caption>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/caption) и этот элемент необходимо поместить внутрь  элемента <table>. Причем вам нужно поместить его сразу после открытия тега <table>.

<table>

 <caption>Dinosaurs in the Jurassic period</caption>

 ...

</table>

Как можно понять из короткого примера выше, заголовок отражает в себе описание контента таблицы. Это полезно для всех читателей просматривающих страницу и желающих получить краткое представление от том полезна ли для них таблица,  что особенно важно для слепых пользователей. Вместо того чтобы читать содержимое множества ячеек чтобы понять о чем таблица, он или она могут полагаться на заголовок и принимать решение читать ли таблицу более подробно.

Заголовок помещают сразу после тега <table>.

**Примечание**: Атрибут summary также может быть использован в <table> элементе предоставляя описание — это также читается скринридерами. Однако мы рекомендуем вместо этого использовать <caption> элемент, так как summary deprecated в HTML5 спецификации и не может быть прочитан зрячими пользователями (он не отображается на странице).

### Упражнение: Добавление заголовка

Давайте попробуем это, вернемся к примеру который мы ранее встретили в прошлой статье.

1. Откройте расписание занятий школьного учителя по языку в конце статьи [HTML таблицы основы](https://developer.mozilla.org/ru/docs/Learn/HTML/Tables/Basics#Active_learning_colgroup_and_col), или сделайте копию нашего [timetable-fixed.html](https://github.com/mdn/learning-area/blob/master/html/tables/basic/timetable-fixed.html) файла.
2. Добвьте подходящий заголовок к таблице.
3. Сохраните свой код и откройте его в браузере, чтобы посмотреть как это выглядит.

**Примечание**: Этот пример можно также найти на GitHub по ссылке [timetable-caption.html](https://github.com/mdn/learning-area/blob/master/html/tables/advanced/timetable-caption.html) ([see it live also](http://mdn.github.io/learning-area/html/tables/advanced/timetable-caption.html)).

## Добавление структуры с помощью <thead>, <tfoot> и <tbody>

Когда таблицы становятся более сложными по структуре полезно дать им более структурированное определение. Отличный способ  сделать это используя <thead>, [<tfoot>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/tfoot) и <tbody>, которые позволяют вам разметить header, footer и body секции таблицы.

Эти элементы не создают дополнительной доступности для пользователей со скринридерами и не приводят к какому-то визуальному улучшению при их использовании. Зато они очень полезны при стилизации и разметке, как точки для добавления CSS к вашей таблице. Вот несколько интересных примеров, в случае длинной таблицы вы можете сделать header и footer таблицы повторяемый на каждой печатной странице, или вы можете сделать body таблицы отбражаемое на одной странице и иметь доступ ко всему содержимому контенту прокручивая вверх и вниз.

Использование:

* Элементом <thead> нужно обернуть часть таблицы которая относится к заголовку — обычно это первая строка содержащая заголовки колонок, но это не обязательно всегда такой случай. Если вы используете [<col>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/col)/<colgroup> элемент, тогда заголовок должен находиться ниже его.
* Элементом <tfoot> нужно обернуть тчасть которая относится к footer таблицы — например, это может быть последняя строка в которой отбражаются суммы по столбцам таблицы. Вы можете включить сюда footer таблицы, как и следовало ожидать, или чуть ниже заголовка таблицы (браузер все равно отобразит его внизу таблицы).
* Элементом <tbody> необходимо обернуть остальную часть содержимого таблицы котор не находится в header или footer таблицы. Этот блок располагают ниже заголовка таблицы или иногда footer таблицы, зависит от того какую структуру вы решите использовать (читать выше по тексту).

**Примечание**: <tbody> всегда включен в каждой таблице, неявно если не укажете его в коде. Проверьте это, открыв один из предыдущих примеров в котором не используется <tbody> и посмотрите HTML код в browser developer tools — вы увидите, что браузер добавил этот тег самостоятельно. Вы могли бы задаться вопросом почему мы должны волноваться о его включениии, но вы должны, потому что это дает больше контроля над структурой таблицы и стилем.

### Упражнение: Добавление структуры таблицы

Давайте используем эти новые элементы.

1. В первую очередь,  сделайте копию [spending-record.html](https://github.com/mdn/learning-area/blob/master/html/tables/advanced/spending-record.html) и [minimal-table.css](https://github.com/mdn/learning-area/blob/master/html/tables/advanced/minimal-table.css) в новой папке.
2. Попробуйте открыть это в браузере — вы увидите, что все выглядит классно, но могло бы быть лучше. Строка "SUM" которая содержит потраченные суммы кажется находится не в том месте и некооторые детали отсутствуют в коде.
3. Поместите очевидную строку заголовка внутрь <thead> элемента, строку "SUM" внутрь <tfoot> элемента и оставшийся контент внутрь <tbody> элемента.
4. Сохраните, презагрузите и вы увидите, что добавление элемента <tfoot> привело к тому,  что строка "SUM" опустилась к нижней части таблицы.
5. Далее, добавьте атрибут [colspan](https://developer.mozilla.org/ru/docs/Web/HTML/Element/td%22%20%5Cl%20%22attr-colspan), чтобы ячейка "SUM" занимала первые четыре столбца, таким образом числовое значение "Cost" появится в последнем столбце.
6. Давайте добавим несколько простых дополнительных стилей к таблице, чтобы дать вам представление насколько эти элементы полезны при использовании CSS. Внутри в <head> вашего HTML документа вы увидите пустой элемент [<style>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/style). Внутри этого элемента добавьте следующие строки CSS кода:
7. tbody {
8. font-size: 90%;
9. font-style: italic;
10. }
11. tfoot {
12. font-weight: bold;
13. }
14. Сохраните, обновите и вы увидите результат. Если <tbody> и <tfoot> элементы не были установлены, то вам придется писать много сложных селекторов/правил для применения одиного и того же стиля.
15. Вложенные таблицы

В одну таблицу вкладывать другую таблицу возможно, если вы используете полную структуру включая элемент <table>. Это как правило не рекомендуется, так как делает разметку более запутанной и менее доступной для пользователей скринридеров, так большинстве случаев вы можете просто вставить дополнительные ячейки/строки/столбцы в существующую таблицу. Однако, иногда это необходимо, например, если вы хотите легко импортировать контент из других источников.

Разметка простой вложенной таблицы:

<table id="table1">

 <tr>

 <th>title1</th>

 <th>title2</th>

 <th>title3</th>

 </tr>

 <tr>

 <td id="nested">

 <table id="table2">

 <tr>

 <td>cell1</td>

 <td>cell2</td>

 <td>cell3</td>

 </tr>

 </table>

 </td>

 <td>cell2</td>

 <td>cell3</td>

 </tr>

 <tr>

 <td>cell4</td>

 <td>cell5</td>

 <td>cell6</td>

 </tr>

</table>

Результат которого выглядит примерно так:

|  |  |  |
| --- | --- | --- |
| **title1** | **title2** | **title3** |
|

|  |  |  |
| --- | --- | --- |
| cell1 | cell2 | cell3 |

 | cell2 | cell3 |
| cell4 | cell5 | cell6 |

## Таблицы для пользователей с ограниченными возможностями

Давайте кратко опишем как мы используем данные таблицы. Таблицы могут быть удобным инструментом, который дает нам быстрый доступ к данным и позволяет искать разные значения. Например, быстрого взгляда на таблицу ниже достаточно, чтобы найти сколько колец было продано в Амстердаме в августе 2016. Чтобы понять эту информацию, мы проводим визуальные ассоциации между данными в этой таблице и ее заголовками колонок и/или строк.

|  |
| --- |
| **Items Sold August 2016** |
|  |  | **Clothes** | **Accessories** |
|  |  | **Trousers** | **Skirts** | **Dresses** | **Bracelets** | **Rings** |
| **Belgium** | **Antwerp** | 56 | 22 | 43 | 72 | 23 |
| **Gent** | 46 | 18 | 50 | 61 | 15 |
| **Brussels** | 51 | 27 | 38 | 69 | 28 |
| **The Netherlands** | **Amsterdam** | 89 | 34 | 69 | 85 | 38 |
| **Utrecht** | 80 | 12 | 43 | 36 | 19 |

Но что если вы не можете провести эти визуальные ассоциации? Как тогда вы сможете прочитать таблицу выше? Люди с ослабленным зрением часто используют скринридер, который читает им информацию с веб-страницы. Это не проблема когда вы читаете простой текст, но интерпретация таблицы может быть сложной проблемой для слепых людей. Тем не менее, вместе с правильной разметкой мы можем заменить визуальные ассоциации програмными.

В этом части статьи приводятся дополнительные способы которые делают таблицы более доступными.

### Использование заголовков столбцов и строк

Скринридеры будут определять все заголовки и использовать их создавая програмные ассоциации между этими заголовками и ячейками к которым они относятся. Сочетание заголовков столбцов и строк будет определять и интерпретировать данные в каждой ячейке так, что пользователи скринридеров могут интерпретировать таблицу также как это делают зрячие пользователи.

Мы уже разобрали заголовки в предыдущей статье, смотри по ссылке [Добавление заголовков с помощью элемента <th>](https://developer.mozilla.org/ru/docs/Learn/HTML/Tables/Basics#Adding_headers_with_%3Cth%3E_elements).

### Атрибут scope

Новая тема в этой статье это атрибут scope, который может быть добавлен к элементу <th> он сообщает скринридеру какие ячейки точно являются заголовками — например, заголовок строки в которой он находится или столбца. Возвращаясь назад к нашему примеру с записями расходов, вы могли однозначно определить заголовки столбцов как здесь:

<thead>

 <tr>

 <th scope="col">Purchase</th>

 <th scope="col">Location</th>

 <th scope="col">Date</th>

 <th scope="col">Evaluation</th>

 <th scope="col">Cost (€)</th>

 </tr>

</thead>

И у каждой строки может быть определен заголовок, как здесь (если мы добавили заголовки строк и заголовки столбцов):

<tr>

 <th scope="row">Haircut</th>

 <td>Hairdresser</td>

 <td>12/09</td>

 <td>Great idea</td>

 <td>30</td>

</tr>

Скринридер распознает разметку структурированную таким образом, что позволяют пользователям прочитать весь столбец или строку целиком.

Атрибут scope имеет два возможных значения — colgroup и rowgroup. Они используются для заголовков, которые располагаются вверху ваших столбцов или строк. Если вы посмотрите на таблицу "Items sold..." в начале этого раздела статьи, вы увидите что ячейка с "Clothes" находится над ячейками "Trousers", "Skirts" и "Dresses". Все эти ячейки должны быть помечены как заголовки (<th>), но "Clothes"  заголовок который находится сверху и определяет остальные три подзаголовка. Поэтому "Clothes" должна получить атрибут scope="colgroup", тогда как другие получат атрибут scope="col".

### Атрибуты id и headers

Альтернатива атрибута scope это использование атрибутов id и [headers](https://developer.mozilla.org/ru/docs/Web/HTML/Element/td%22%20%5Cl%20%22attr-headers) задавая ассоциации между заголовками и ячейками. Этот способ выглядит следующим образом:

1. Вы устанавливаете уникальный id для каждого<th> элемента.
2. Вы устанавливаете атрибут headers для каждого <td> элемента. Каждый headers атрибут должен содержать список всех id , разделенный пробелами, ко всем <th> элементам которые действуют как заголовок для этой ячейки.

Это обеспечивает явное определение позиции для каждой ячейки вашей HTML таблицы, определяет заголовоки столбцов и строк таблицы. Для того чтобы это работало реально хорошо таблице нужно определить и заголовки столбцов, и заголовки строк.

Вернемся к нашеву примеру с расчетом затрат, его можно переписать следующим образом:

<thead>

 <tr>

 <th id="purchase">Purchase</th>

 <th id="location">Location</th>

 <th id="date">Date</th>

 <th id="evaluation">Evaluation</th>

 <th id="cost">Cost (€)</th>

 </tr>

</thead>

<tbody>

<tr>

 <th id="haircut">Haircut</th>

 <td headers="location haircut">Hairdresser</td>

 <td headers="date haircut">12/09</td>

 <td headers="evaluation haircut">Great idea</td>

 <td headers="cost haircut">30</td>

</tr>

 ...

</tbody>

**Примечание**: Этот метод создания очень точного определения ассоциаций между заголовками и данными в ячейках, но использует **гораздо** больше разметки и оставляет обширное пространство для ошибок.  Атрибута scope обычно достаточно для большиства таблиц.

### Упражнение: играем со scope и headers

1. Для заключительного упражнения мы, вначале создадим копию [items-sold.html](https://github.com/mdn/learning-area/blob/master/html/tables/advanced/items-sold.html) и [minimal-table.css](https://github.com/mdn/learning-area/blob/master/html/tables/advanced/minimal-table.css) в новой папке.
2. Теперь попробуем добавить соответствующий атрибут scope, который наиболее соответствует этой таблице.
3. И наконец попробуем сделать другую копию изначальных файлов, на этот раз делая таблицу более доступной используя атрибуты id и headers.

 Заключение

Есть еще некоторые вещи которые можно узнать о HTML таблицах, но мы действительно дали все что нужно на настоящий момент.