

## Рецепт 7. Стилизовое оформление таблиц с использованием псевдоклассов

*Псевдоклассы* CSS предназначены для выбора элементов на основании информации, не входящей в документ, или информации, которая не может быть выражена обычными селекторами. Вероятно, вы уже использовали псевдоклассы ранее — например, `:hover` для изменения цвета ссылки, когда пользователь наводит на нее указатель мыши. В CSS3 появилось несколько новых псевдоклассов, заметно упрощающих поиск элементов.

### Работа со счетами

Фирма AwesomeCo использует систему выставления счетов для продаваемых товаров. Одним из основных рынков AwesomeCo является корпоративная сувенирная продукция — ручки, чашки, футболки и вообще все, на что можно «налепить» фирменный логотип. Вам поручено сделать работу со счетами более простой и удобной. В существующей версии системы разработчики создают стандартную таблицу HTML, примерный вид которой показан на рис. 4.1.

Item	Price	Quantity	Total
Coffee mug	\$10.00	5	\$50.00
Polo shirt	\$20.00	5	\$100.00
Red stapler	\$9.00	4	\$36.00
Subtotal			\$186.00
Shipping			\$12.00
Total Due			\$198.00

**Рис. 4.1.** Таблица HTML без стилизового оформления в текущей версии системы

Перед вами довольно стандартный счет с ценами, количеством единиц товара, суммами строк, промежуточной суммой и общей суммой заказа. Счет было бы удобнее просматривать, если бы строки были окрашены в разные цвета. Также будет полезно изменить цвет итоговой суммы, чтобы она лучше выделялась на общем фоне.

Ниже приведен код таблицы. Скопируйте его в отдельный файл, чтобы с ним было удобнее работать.

## css3advancedselectors/table.html

```
<table >
  <tr>
    <th>Item</th>
    <th>Price</th>
    <th>Quantity</th>
    <th>Total</th>
  </tr>
  <tr>
    <td>Coffee mug</td>
    <td>$10.00</td>
    <td>5</td>
    <td>$50.00</td>
  </tr>
  <tr>
    <td>Polo shirt</td>
    <td>$20.00</td>
    <td>5</td>
    <td>$100.00</td>
  </tr>
  <tr>
    <td>Red stapler</td>
    <td>$9.00</td>
    <td>4</td>
    <td>$36.00</td>
  </tr>
  <tr>
    <td colspan="3">Subtotal</td>
    <td>$186.00</td>
  </tr>
  <tr>
    <td colspan="3">Shipping</td>
    <td>$12.00</td>
  </tr>
  <tr>
    <td colspan="3">Total Due</td>
    <td>$198.00</td>
  </tr>
</table>
```

Прежде всего избавимся от уродливой стандартной сетки.

**css3advancedselectors/table.html**

```
table{
  width: 600px;
  border-collapse: collapse;
}
th, td{
  border: none;
}
```

Также немного изменим оформление заголовка: он будет выводиться белыми буквами на черном фоне.

**css3advancedselectors/table.html**

```
th{
  background-color: #000;
  color: #fff;
}
```

После применения стиля таблица выглядит так:

Item	Price	Quantity	Total
Coffee mug	\$10.00	5	\$50.00
Polo shirt	\$20.00	5	\$100.00
Red stapler	\$9.00	4	\$36.00
Subtotal			\$186.00
Shipping			\$12.00
Total Due			\$198.00

После удаления сетки и увеличения ширины можно переходить к стилизованному оформлению отдельных строк и столбцов с использованием псевдоклассов. Начнем с чередования цвета строк.

## Чередование цвета строк (:nth-of-type)

Каждый из нас неоднократно видел таблицы с чередованием цвета строк («зебра»): этот эффект полезен тем, что упрощает просмотр данных по строкам. Стилизованное оформление такого рода лучше всего выполняется средствами уровня представления, то есть CSS. Традиционно задача решалась включением в строки таблицы дополнительных имен классов (например, `odd` и `even` для нечетных и четных строк соответственно).

Однако подобное загрязнение разметки таблицы нежелательно, поскольку спецификация HTML5 рекомендует избегать использования имен классов для определения представления. При помощи новых селекторов мы сможем добиться желаемого эффекта без изменения разметки — таким образом, представление будет отделено от контента.

Селектор `nth-of-type` находит каждый  $n$ -й элемент конкретного типа, определяемый формулой или ключевыми словами. Формулы будут более подробно рассмотрены позднее, а пока разберемся с ключевыми словами, потому что их проще понять.

Чтобы каждая вторая строка таблицы была окрашена в другой цвет, проще всего найти все четные строки таблицы и назначить им другой цвет фона. То же самое делается с нечетными строками. В CSS3 имеются ключевые слова `even` и `odd`, предназначенные именно для таких ситуаций.

#### css3advancedselectors/table.html

```
tr:nth-of-type(even){
  background-color: #F3F3F3;
}
tr:nth-of-type(odd) {
  background-color:#ddd;
}
```

Фактически этот селектор означает: «Найти каждую четную строку таблицы и задать ее цвет. Затем найти каждую нечетную строку таблицы и задать ее цвет». Так «зебровая» окраска таблицы реализуется без использования сценарного кода или дополнительных имен классов в строках.

Очередная версия стилового оформления таблицы выглядит так:

Item	Price	Quantity	Total
Coffee mug	\$10.00	5	\$50.00
Polo shirt	\$20.00	5	\$100.00
Red stapler	\$9.00	4	\$36.00
Subtotal			\$186.00
Shipping			\$12.00
Total Due			\$198.00

Теперь поработаем над выравниванием столбцов в таблице.

## Выравнивание текста столбцов (:nth-child)

По умолчанию текст во всех столбцах таблицы выравнивается по левому краю. Мы выровняем по правому краю все столбцы, кроме первого, — чтобы цена и количество единиц товара лучше читались. Для этого мы воспользуемся селектором `nth-child`, но сначала необходимо узнать, как он работает.

Селектор `nth-child` ищет дочерние элементы заданного элемента; по аналогии с `nth-of-type`, он может использовать ключевые слова или формулу.

Формула определяется в виде `an+b`, где **a** — множитель, а **b** — смещение. Принцип использования формул проще понять в контексте; давайте применим его к контексту таблицы.

Для выбора всех строк таблицы можно воспользоваться селектором вида

```
table tr:nth-child(n)
```

В этом примере не указан ни множитель, ни смещение.

Все строки таблицы, кроме первой (строка с заголовками столбцов), выбираются при помощи селектора со смещением:

```
table tr:nth-child(n+2)
```

А для выбора каждой второй строки таблицы используется множитель `2n`:

```
table tr:nth-child(2n)
```

Каждая третья строка выбирается при помощи множителя `3n`.

Если прибавить к множителю смещение, то поиск будет начинаться не от начала таблицы, а с одной из следующих строк. Следующий селектор находит каждую вторую строку, начиная с четвертой:

```
table tr:nth-child(2n+4)
```

Итак, для выравнивания всех столбцов, *кроме* первого, используется следующая запись:

```
css3advancedselectors/table.html
```

```
td:nth-child(n+2){
  text-align: right;
}
```

Наша таблица постепенно приобретает все более профессиональный вид.

Item	Price	Quantity	Total
Coffee mug	\$10.00	5	\$50.00
Polo shirt	\$20.00	5	\$100.00
Red stapler	\$9.00	4	\$36.00
Subtotal			\$186.00
Shipping			\$12.00
Total Due			\$198.00

На следующем этапе мы изменим оформление последней строки таблицы.

## Выделение последней строки (:last-child)

Таблица уже сейчас смотрится вполне прилично, но начальство требует, чтобы нижняя строка выделялась жирным шрифтом. Для этого мы воспользуемся селектором `last-child`, который находит последний дочерний элемент группы.

Многие веб-разработчики устанавливают нижние поля в абзацах для более равномерного распределения текста по странице. Однако иногда это приводит к образованию нежелательных интервалов в конце группы. Например, если абзацы содержатся в боковой панели или на выноске, нижние поля из последнего абзаца желательно исключить, чтобы он не отделялся от границы области. Селектор `last-child` идеально подходит для отмены полей последнего абзаца.

```
p{ margin-bottom: 20px }
#sidebar p:last-child{ margin-bottom: 0; }
```

Аналогичным образом текст нижней строки выделяется жирным шрифтом.

### css3advancedselectors/table.html

```
tr:last-child{
  font-weight: bolder;
}
```

Выделение также применяется и к последнему столбцу таблицы, чтобы суммы строк тоже выделялись на общем фоне:

**css3advancedselectors/table.html**

```
td:last-child{
  font-weight: bolder;
}
```

Наконец, селектор `last-child` используется для увеличения размера шрифта общей суммы в правом нижнем углу таблицы. Мы находим последний столбец последней строки и изменяем его оформление.

**css3advancedselectors/table.html**

```
tr:last-child td:last-child{
  font-size:24px;
}
```

Item	Price	Quantity	Total
Coffee mug	\$10.00	5	\$50.00
Polo shirt	\$20.00	5	\$100.00
Red stapler	\$9.00	4	\$36.00
Subtotal			\$186.00
Shipping			\$12.00
<b>Total Due</b>			<b>\$198.00</b>

Работа почти закончена. Осталось лишь кое-что сделать с тремя последними строками таблицы.

## Поиск в обратном направлении (`:nth-last-child`)

Если стоимость доставки снижена под действием скидки, то соответствующая строка таблицы должна выделяться цветом. Для быстрого поиска этой строки удобно использовать селектор `nth-last-child`. Вы уже видели, как селектор `nth-child` и формула `a+n` используются для выбора конкретных дочерних элементов (см. раздел «Выравнивание текста столбцов (`:nth-child`)» этой главы). Селектор `nth-last-child` работает практически так же, если не считать того, что он перебирает дочерние элементы в обратном порядке, начиная с последнего. Это позволяет легко найти предпоследний элемент группы, что, собственно, и нужно сделать в нашем примере.

Итак, оформление строки со стоимостью доставки может быть изменено следующим кодом:

**css3advancedselectors/table.html**

```
tr:nth-last-child(2){
  color: green;
}
```

Селектор определяет конкретный дочерний элемент — второй с конца.

В оформление таблицы осталось внести последний штрих. Ранее мы выравнивали по правому краю все столбцы, кроме первого. Для строк с описаниями и ценами товаров такое выравнивание естественно, но последние три строки выглядят немного странно. Для них лучше использовать выравнивание по правому краю. Для решения этой задачи мы используем селектор `nth-last-child` с отрицательным множителем и положительным смещением.

**css3advancedselectors/table.html**

```
tr:nth-last-child(-n+3) td{
  text-align: right;
}
```

Такая формула реализует интервальный выбор. В ней используется смещение 3, а с селектором `nth-last-child` выбирается каждый элемент до заданного смещения. Если бы вместо него использовался селектор `nth-child`, то строки выбирались бы от начала таблицы.

Новое оформление таблицы (рис. 4.2) выглядит намного лучше, причем нам совершенно не пришлось изменять разметку. Многие селекторы, использованные в решении, пока недоступны пользователям Internet Explorer; для них нужно разработать обходное решение.

Item	Price	Quantity	Total
Coffee mug	\$10.00	5	<b>\$50.00</b>
Polo shirt	\$20.00	5	<b>\$100.00</b>
Red stapler	\$9.00	4	<b>\$36.00</b>
		Subtotal	<b>\$186.00</b>
		Shipping	<b>\$12.00</b>
		<b>Total Due</b>	<b>\$198.00</b>

**Рис. 4.2.** Таблица с чередованием цветов и выравниванием, реализованным исключительно средствами CSS3



## Обходное решение

Селекторы, использованные в нашем решении, поддерживаются текущими версиями Opera, Firefox, Safari и Chrome, но в Internet Explorer 8.0 и более ранних версий они игнорируются. Необходимо реализовать обходное решение, причем разработчик сталкивается с принципиальным выбором.

### Изменение кода HTML

Самое очевидное решение, которое будет работать во всех браузерах, — модификация базового кода. Ко всем ячейкам таблицы присоединяются классы, а для каждого класса задается базовое оформление CSS. Смещение представления с контентом — самый худший вариант; ведь именно для предотвращения подобной мешанины используется CSS3. В будущем вся эта разметка станет лишней, а ее удаление создаст изрядные трудности.

### Использование JavaScript

Библиотека jQuery уже поддерживает большинство селекторов CSS3, поэтому написать нужный метод стиливого оформления таблицы несложно, но существует и более простое решение.

Кейт Кларк написал отличную библиотеку IE-css3<sup>1</sup>, которая реализует поддержку селекторов CSS3 в Internet Explorer. Для этого необходимо лишь включить пару сценариев в нашу страницу.

Библиотека IE-CSS3 в своей внутренней реализации может использовать jQuery, Prototype или несколько других библиотек, но я предпочитаю библиотеку DOMAssistant<sup>2</sup>, потому что в ней реализована наиболее качественная поддержка псевдоклассов.

Загрузите обе библиотеки и подключите их к своему документу. Так как это решение предназначено только для IE, поместите их в условный комментарий, чтобы библиотеки могли использоваться только пользователями IE.

```
css3advancedselectors/table.html
```

```
<!--[if (gte IE 5.5)&(lte IE 8)]>
  <script type="text/javascript">
```

<sup>1</sup> <http://www.keithclark.co.uk/labs/ie-css3/>

<sup>2</sup> <http://www.domassistant.com/>

```

        src="js/DOMAssistantCompressed-2.8.js"></script>
<script type="text/javascript"
        src="js/ie-css3.js"></script>
<![endif]-->

```

Включение этих сценариев в страницу решает все проблемы с оформлением таблицы в Internet Explorer. Результат показан на рис. 4.3.



Item	Price	Quantity	Total
Coffee mug	\$10.00	5	\$50.00
Polo shirt	\$20.00	5	\$100.00
Red stapler	\$9.00	4	\$36.00
		Subtotal	\$186.00
		Shipping	\$12.00
		<b>Total Due</b>	<b>\$198.00</b>

**Рис. 4.3.** Таблица нормально воспроизводится в Internet Explorer

Решение работает только при включенной поддержке JavaScript, но стилевое оформление предназначено в основном для удобства просмотра. По крайней мере, его отсутствие не мешает пользователю просмотреть содержимое счета.

CSS3 значительно упрощает стилевое оформление элементов, особенно если вы не можете внести изменения в целевой код HTML. Занимаясь стилевым оформлением интерфейсов, по возможности используйте семантическую иерархию и новые селекторы без внесения дополнительной разметки — это существенно упростит сопровождение вашего кода.