

**Геннадий Самков**

# **jQuery**

## **Сборник рецептов**

Санкт-Петербург

«БХВ-Петербург»

2010

УДК 681.3.068+800.92 jQuery  
ББК 32.973.26-018.1  
С17

**Самков Г. А.**

С17 jQuery. Сборник рецептов. — СПб.: БХВ-Петербург, 2010. — 416 с.: ил. + CD-ROM — (Профессиональное программирование)  
ISBN 978-5-9775-0495-9

Книга является сборником решений наиболее часто встречающихся задач при веб-программировании пользовательских интерфейсов с использованием библиотеки jQuery. Рассмотрены практически все методы и вспомогательные функции jQuery, в том числе обеспечивающие взаимодействие jQuery и AJAX. Подробно рассказано о настройке UI jQuery и приведены описания всех настроек для виджетов, входящих в ее состав, что позволяет использовать книгу в качестве справочника. Приведено большое количество примеров использования наиболее популярных плагинов для jQuery — создание графиков и диаграмм, фотогалерей, навигационных меню, всплывающих подсказок, работа с веб-формами, таймерами и cookies, обработка табличных данных. Компакт-диск содержит примеры, разобранные в книге, файлы библиотеки jQuery версий 1.2.6 и 1.3.2, файлы настройки UI jQuery, а также файлы рассмотренных в книге расширений сторонних разработчиков.

*Для веб-программистов*

УДК 681.3.068+800.92 jQuery  
ББК 32.973.26-018.1

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Леонид Кочин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 03.12.09.  
Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 33,54.  
Тираж 1500 экз. Заказ №  
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию  
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой  
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0495-9

© Самков Г. А., 2009  
© Оформление, издательство "БХВ-Петербург", 2009

# Оглавление

<b>Введение .....</b>	<b>1</b>
Структура книги .....	1
Как работать с книгой.....	2
Источники информации .....	3
Благодарности .....	4
<b>ЧАСТЬ I. МЕТОДЫ БИБЛИОТЕКИ JQUERY .....</b>	<b>5</b>
<b>Глава 1. Выбор элементов .....</b>	<b>7</b>
1.1. Базовые правила .....	7
1.2. Выбор элементов с учетом иерархии .....	14
1.3. Основные фильтры .....	18
1.4. Фильтрация по содержимому .....	24
1.5. Фильтры видимых и невидимых элементов .....	28
1.6. Фильтры атрибутов.....	29
1.7. Фильтры элементов форм .....	35
1.8. Фильтры состояния элементов форм .....	37
1.9. Фильтры элементов-потомков .....	42
<b>Глава 2. Атрибуты элементов.....</b>	<b>48</b>
2.1. Управление атрибутами элементов .....	48
2.2. Работа с атрибутом <i>class</i> .....	53
2.3. Работа с HTML и текстом .....	54
2.4. Работа с атрибутом <i>value</i> .....	57
<b>Глава 3. Визуальные эффекты .....</b>	<b>64</b>
3.1. Как показывать и скрывать элементы .....	64
3.2. Эффекты "скольжения" и "затухания" .....	67
3.3. Создание анимации.....	71
3.4. Эффекты UI jQuery .....	76

<b>Глава 4. Работа с CSS-свойствами</b> .....	<b>81</b>
4.1. Как получать и устанавливать значения CSS-свойств элементов.....	81
4.2. Ширина и высота элементов.....	85
4.3. Позиционирование элементов.....	88
<b>Глава 5. Некоторые методы ядра библиотеки jQuery</b> .....	<b>92</b>
5.1. Примеры работы с объектом jQuery.....	92
5.2. Сохранение и извлечение данных.....	97
<b>Глава 6. Манипуляции над элементами</b> .....	<b>101</b>
6.1. Изменение содержимого элементов.....	101
6.2. Как вставлять элементы в DOM.....	103
6.3. Замена, удаление и копирование элементов.....	113
<b>Глава 7. Перемещение по элементам</b> .....	<b>120</b>
7.1. Поиск нужных элементов в DOM.....	120
7.2. Фильтрация элементов набора.....	131
7.3. Перемещение по цепочке вызовов.....	138
<b>Глава 8. События и их обработка</b> .....	<b>142</b>
8.1. Готовность документа.....	142
8.2. Назначение, удаление и вызов событий.....	144
8.3. Взаимодействие с элементами.....	152
8.4. События.....	155
<b>Глава 9. Взаимодействие jQuery и AJAX</b> .....	<b>161</b>
9.1. Самое простое.....	161
9.2. GET- и POST-запросы.....	166
9.3. Вспомогательная функция <i>\$.ajax(options)</i> .....	174
9.4. Для чего нужна функция <i>\$.ajaxSetup(options)</i> .....	180
9.5. События AJAX.....	183
<b>Глава 10. Утилиты jQuery</b> .....	<b>190</b>
10.1. Некоторые операции с массивами и объектами в jQuery.....	190
<b>ЧАСТЬ II. РАСШИРЕНИЯ ДЛЯ БИБЛИОТЕКИ JQUERY</b> .....	<b>201</b>
<b>Глава 11. Меню для веб-сайта</b> .....	<b>203</b>
11.1. Плагин jQuery Multi Level Menu.....	203
11.2. Плагин jQuery Drop Line Menu.....	207
11.3. Плагин jQuery TreeView.....	210
11.4. Делаем меню похожее на Accordion.....	216

<b>Глава 12. Работа с таблицами</b> .....	<b>219</b>
12.1. Плагин jQuery DataTables.....	219
<b>Глава 13. Графики и диаграммы</b> .....	<b>229</b>
13.1. Плагин jqPlot.....	229
<b>Глава 14. AJAX-формы</b> .....	<b>242</b>
14.1. Плагин jQuery Autocomplete.....	242
14.2. Плагин jQuery Form.....	252
14.3. Плагин jQuery Validate.....	257
14.4. Плагин jQuery Uploadify.....	264
<b>Глава 15. Фотогалереи для сайта</b> .....	<b>274</b>
15.1. Фотогалерея FancyBox.....	274
15.2. Простая фотогалерея.....	282
<b>Глава 16. Несколько полезных плагинов</b> .....	<b>286</b>
16.1. jQuery Cookie.....	286
16.2. jQuery Corner.....	288
16.3. jQuery Timers.....	293
16.4. jQuery Cluetip.....	297
<b>Глава 17. UI jQuery — виджеты</b> .....	<b>303</b>
17.1. Виджет Accordion.....	303
17.2. Виджет DatePicker.....	314
17.3. Виджет Dialog.....	326
17.4. Виджет Progressbar.....	334
17.5. Виджет Slider.....	337
17.6. Виджет Tabs.....	343
<b>Глава 18. UI jQuery — взаимодействие с элементами страницы</b> .....	<b>354</b>
18.1. Draggable — перемещение элементов.....	354
18.2. Droppable — "сброс" элементов.....	364
18.3. Resizable — изменение размеров элементов.....	371
18.4. Selectable — выбор элементов.....	377
18.5. Sortable — сортировка элементов.....	385
<b>Литература</b> .....	<b>397</b>
<b>Приложение. Описание компакт-диска</b> .....	<b>398</b>
<b>Предметный указатель</b> .....	<b>400</b>



# Введение

Предлагаемая книга является сборником решений наиболее часто встречающихся задач при программировании пользовательских интерфейсов с использованием JavaScript-библиотеки jQuery.

Предполагается, что читатель знаком с основами JavaScript, CSS и HTML. Кроме того, потребуются некоторые знания основ PHP — языка программирования серверных сценариев, который необходим в некоторых примерах, посвященных организации взаимодействия клиент — сервер с применением технологии AJAX.

Книга может служить не только учебником, но и справочником по библиотеке jQuery и надстройке UI jQuery.

## Структура книги

Книга содержит две части и приложение.

В *части I* решения задач представлены так, чтобы помочь читателю на простых примерах освоить подавляющее большинство методов библиотеки jQuery, имеющихся в распоряжении разработчика. Подробно освещены такие вопросы, как:

- выбор и фильтрация элементов;
- работа с атрибутами элементов;
- создание визуальных эффектов, в том числе с использованием возможностей надстройки UI jQuery;
- работа с CSS-свойствами элементов;
- организация доступа к свойствам объекта jQuery;
- манипуляции элементами: изменение содержимого, добавление, замена, удаление и копирование элементов;

- ❑ перемещение по элементам объектной модели документа;
- ❑ работа с событиями (назначение, удаление и вызов событий);
- ❑ взаимодействие jQuery и AJAX;
- ❑ некоторые полезные утилиты библиотеки jQuery.

В *части II* приведены решения на основе наиболее популярных расширений для библиотеки jQuery, в том числе рассмотрен официальный пакет расширений UI jQuery. Подробно рассматриваются:

- ❑ меню для веб-сайтов (многоуровневые меню, меню на основе плагина TreeView, а также меню, похожее на популярный виджет Accordion);
- ❑ организация работы с данными, представленными в табличной форме, рассматривается на примере плагина DataTables;
- ❑ возможности реализации графиков и диаграмм на страницах веб-сайта демонстрируются на примере плагина jqPlot;
- ❑ работа с AJAX-формами — плагины Autocomplete (автоматические подсказки в форме), Form (организация AJAX-формы), Validate (проверка данных в AJAX-форме) и FileUpload (загрузка файлов на сервер);
- ❑ фотогалереи для веб-сайтов рассматриваются на примере плагина FancyBox, кроме того, приведено решение по созданию простейшей встроенной фотогалереи из нескольких строк jQuery-кода;
- ❑ некоторые полезные плагины — плагин jQuery Corner (уголки без использования графики), плагин jQuery Cookie (установка и считывание cookie), плагин jQuery ClueTip (всплывающие подсказки), плагин jQuery Timers (управление таймерами);
- ❑ виджеты надстройки UI jQuery — Accordion (раскрывающееся меню), DatePicker (выбор даты), Dialog — (диалоговое окно), ProgressBar (шкала загрузки), Slider (шкала с бегунком) и Tabs (организация переключения вкладок);
- ❑ надстройка UI jQuery — взаимодействие с элементами страницы: Draggable (перемещение элементов), Droppable ("сброс" элементов), Resizable (изменение размеров элементов), Selectable (выбор элементов), Sortable (сортировка элементов).

В *приложении* описан компакт-диск, прилагаемый к книге.

## Как работать с книгой

Книга в основном ориентирована на разработчика, располагающего компьютером с операционной системой Windows, но пользователь UNIX также сможет выполнить на своем компьютере все примеры.



В ходе чтения следует выполнять на своем компьютере все примеры, описываемые в книге. Рекомендуем читателю самостоятельно изменять и переделывать каждый пример, чтобы лучше понять, как он работает.

Автор приложил все усилия, чтобы изложить материал с наибольшей точностью, но не исключает возможности ошибок и опечаток. Автор также не несет ответственности за последствия использования сведений, изложенных в книге.

## Источники информации

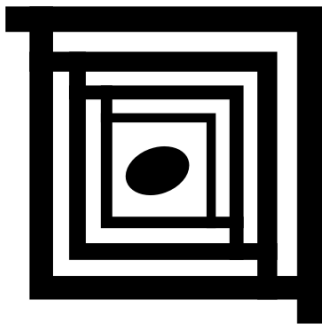
В книге невозможно охватить все вопросы, и читателю наверняка потребуются дополнительные сведения, например, из сети Интернет. Кроме того, могут изменяться версии программного обеспечения, рассматриваемого в книге. Вот адреса, которыми вы можете воспользоваться:

- ☐ <http://jquery.com> — официальный сайт библиотеки jQuery (англ.);
- ☐ [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page) — оригинальная документация по библиотеке jQuery (англ.);
- ☐ <http://slyweb.ru/jquerymain/> — перевод документации jQuery (рус.);
- ☐ <http://jquery-docs.ru> — перевод документации jQuery (рус.);
- ☐ <http://jqueryui.com> — официальный сайт надстройки UI jQuery (англ.);
- ☐ <http://www.linkexchanger.su> — блог автора книги, содержит много статей с примерами использования библиотеки jQuery (рус.);
- ☐ <http://plugins.jquery.com/> — на сайте представлено большое количество разнообразных плагинов для библиотеки jQuery (англ.);
- ☐ <http://bassistance.de/jquery-plugins/jquery-plugin-treeview/> — страница плагина jQuery TreeView (англ.);
- ☐ <http://www.datatables.net> — сайт плагина jQuery DataTables (англ.);
- ☐ <http://www.jqplot.com> — сайт плагина jqPlot (англ.);
- ☐ <http://bassistance.de/jquery-plugins/jquery-plugin-autocomplete/> — страница плагина jQuery Autocomplete (англ.);
- ☐ <http://malsup.com/jquery/form/> — сайт плагина jQuery Form (англ.);
- ☐ <http://bassistance.de/jquery-plugins/jquery-plugin-validation/> — страница плагина jQuery Validation (англ.);
- ☐ <http://www.uploadify.com> — сайт плагина jQuery Uploadify (англ.);
- ☐ <http://fancybox.net> — сайт плагина jQuery FancyBox (англ.);

- ❑ <http://plugins.learningjquery.com/cluetip/> — страница плагина jQuery ClueTip (англ.);
- ❑ <http://www.stilbuero.de/2006/09/17/cookie-plugin-for-jquery/> — страница плагина jQuery Cookie (англ.);
- ❑ <http://jquery.malsup.com/corner/> — страница плагина jQuery Corner (англ.);
- ❑ <http://jquery.offput.ca/timers/> — страница плагина jQuery Timers (англ.);
- ❑ <http://firebug.ru> — сайт, посвященный Firebug, замечательному средству отладки JavaScript-кода (рус.).

## Благодарности

Автор приносит свою благодарность преподавателю компьютерных дисциплин Санкт-Петербургского государственного политехнического университета Елене Сергеевне Бенкен за помощь и поддержку.



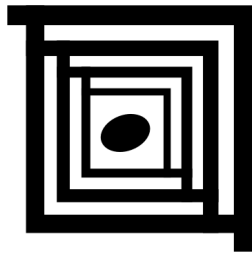
# ЧАСТЬ I

## Методы библиотеки jQuery

<b>Глава 1.</b>	Выбор элементов
<b>Глава 2.</b>	Атрибуты элементов
<b>Глава 3.</b>	Визуальные эффекты
<b>Глава 4.</b>	Работа с CSS-свойствами
<b>Глава 5.</b>	Некоторые методы ядра библиотеки jQuery
<b>Глава 6.</b>	Манипуляции над элементами
<b>Глава 7.</b>	Перемещение по элементам
<b>Глава 8.</b>	События и их обработка
<b>Глава 9.</b>	Взаимодействие jQuery и AJAX
<b>Глава 10.</b>	Утилиты jQuery



# ГЛАВА 1



## Выбор элементов

А для чего вообще нужно выбирать элементы? Ответ простой — для того, чтобы как-то на них воздействовать. Например, можно изменить атрибуты или CSS-свойства элементов, меняя, таким образом, их визуальное представление, изменять содержимое этих элементов, связывать с ними определенные события и т. д.

Можно выбрать как один элемент, так и множество элементов. Независимо от того, сколько именно элементов будет выбрано, мы будем рассматривать это как набор элементов, называя его объектом jQuery.

### 1.1. Базовые правила

#### Проблема

Необходимо отыскать абсолютно все элементы веб-страницы.

#### Решение

Используем селектор `*` для решения этой задачи (листинг 1.1.1).

##### Листинг 1.1.1. Использование селектора `*`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-1-1</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    alert($(".*").length);
});
</script>
</head>
<body>
<ul>
    <li></li>
    <li></li>
</ul>
<p></p>
<div><span></span></div>
</body>
</html>
```

## Обсуждение

Чтобы рассмотренный пример не выглядел совсем скучно, и можно было понять, что он действительно работает, мы не только выбрали в объект jQuery все элементы веб-страницы, но также подсчитали их число и вывели его в окне предупреждения. Поскольку контекстом в приведенном примере является объект document, то в набор попадут элементы не только из body, но и из head. В наборе также окажутся элементы script и т. п.

## Проблема

Необходимо отыскать все элементы веб-страницы, но только в контексте body, исключив элементы, входящие в head.

## Решение

Для решения этой задачи так же воспользуемся селектором \*, но в качестве второго аргумента явно передадим контекст (листинг 1.1.2).

### Листинг 1.1.2. Использование селектора \*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
```

```
<title>example-1-1-2</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    alert($("#*", document.body).length);
});
</script>
</head>
<body>
<ul>
    <li></li>
    <li></li>
</ul>
<p></p>
<div><span></span></div>
</body>
</html>
```

## Обсуждение

Мы точно так же вывели в окно предупреждения число выбранных элементов. Заметили разницу? 6 против 14 в примере из листинга 1.1.1. В набор не попали элементы из head.

## Проблема

Необходимо отыскать элемент по известному значению его атрибута id.

## Решение

Для поиска элемента по значению его атрибута id воспользуемся селектором идентификатора (листинг 1.1.3).

### Листинг 1.1.3. Использование селектора #id

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-1-3</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
```

```
<script type="text/javascript">
$(function() {
    $("#myDiv").css("border", "1px solid #f00");
});
</script>
<style type="text/css">
div { width:150px; height:150px; border:1px solid #00f; margin:2px; }
</style>
</head>
<body>
<div class="myDiv"></div>
<div></div>
<div id="myDiv"></div>
<div><span></span></div>
<div id="otherDiv"></div>
</body>
</html>
```

## Обсуждение

Отыскав элемент, который имеет значение идентификатора `myId`, мы применили к нему метод `css()`, добавив выбранному элементу красную рамку шириной в 1 px, чтобы убедиться в том, что селектор действительно обнаружил нужный элемент.

## Проблема

Необходимо отыскать элемент по значению атрибута `id`, в который входят специфические символы, такие как точка или квадратные скобки. Проблема состоит в том, что эти символы имеют специальное значение в CSS.

## Решение

Снова воспользуемся селектором идентификатора, но перед специальными символами поставим два обратных слэша подряд (листинг 1.1.4).

### Листинг 1.1.4. Использование селектора `#id`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-1-4</title>
```



```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    $("#my\\.Div").css("border", "1px solid #f00");
    $("#my\\[Div\\]").css("border", "1px solid #0f0");
});
</script>
<style type="text/css">
div { width:150px; height:150px; border:1px solid #00f; margin:2px; }
</style>
</head>
<body>
<div></div>
<div id="my.Div"></div>
<div><span></span></div>
<div id="my[Div]"></div>
</body>
</html>
```

## Обсуждение

Чтобы убедиться, что этот прием работает корректно, мы с помощью метода `css()` устанавливаем для найденных элементов различный цвет рамок.

## Проблема

Необходимо отыскать все элементы определенного типа, например, все элементы `div` на веб-странице.

## Решение

Для решения задачи нужно всего лишь воспользоваться селектором `element` (листинг 1.1.5).

### Листинг 1.1.5. Использование селектора `element`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-1-5</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    $("div").css("border", "1px solid #f00");
});
</script>
<style type="text/css">
div { width:150px; height:150px; border:1px solid #00f; margin:2px; }
p { width:150px; height:150px; border:1px solid #00f; margin:2px; }
</style>
</head>
<body>
<div></div>
<p></p>
<div></div>
<p></p>
<div></div>
</body>
</html>
```

## Обсуждение

С помощью селектора `element` нам удалось выбрать все элементы `div`, имеющиеся на веб-странице. Используя метод `css()`, мы установили для выбранных элементов рамки красного цвета шириной в 1 px.

## Проблема

Необходимо отыскать элемент (или элементы) по имени класса.

## Решение

Для решения задачи применяется селектор, который в точности повторяет синтаксис `css` (листинг 1.1.6).

### Листинг 1.1.6. Использование селектора `.class`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-1-6</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    $(".test").css("border", "1px solid #f00");
});
</script>
<style type="text/css">
div, p, ul { border:1px solid #00f; margin:2px; }
</style>
</head>
<body>
<div>div</div>
<p class="test">p class="test"</p>
<ul class="test">
    <li>li списка ul class="test"</li>
    <li>li списка ul class="test"</li>
</ul>
<p>p</p>
<div class="test">div class="test"</div>
</body>
</html>
```

## Обсуждение

С помощью селектора `.class` мы выбрали все элементы, которые имеют значение `test` атрибута `class`. Для наглядности вновь используем метод `css()`, чтобы установить для выбранных элементов рамки красного цвета шириной в 1 px.

## Проблема

Необходимо отыскать элементы, которые имеют сразу несколько имен классов.

## Решение

Для решения задачи применяется селектор `.class.class` (листинг 1.1.7).

### Листинг 1.1.7. Использование селектора `.class.class`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
```

```
<head>
<title>example-1-1-7</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    $(".oneTest.twoTest").css("border", "1px solid #f00");
});
</script>
<style type="text/css">
div { border:1px solid #00f; margin:2px; }
</style>
</head>
<body>
<div class="oneTest">div class="oneTest"</div>
<div class="twoTest">div class="twoTest"</div>
<div class="oneTest twoTest">div class="oneTest twoTest"</div>
</body>
</html>
```

## Обсуждение

С помощью селектора `.class.class` мы выбрали элемент, который имеет сразу оба названия класса в атрибуте `class`. Опять используем метод `css()`, чтобы установить для выбранных элементов рамки красного цвета шириной в 1 px.

## 1.2. Выбор элементов с учетом иерархии

### Проблема

Необходимо отыскать элементы, являющиеся потомками какого-либо элемента.

### Решение

Используем селектор `ancestor descendant` для решения этой задачи (листинг 1.2.1).

**Листинг 1.2.1. Использование селектора ancestor descendant**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-2-1</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<style type="text/css">
form {
    border:2px green solid;
    padding:2px;
    margin:0;
    background:#efe;
}
div {
    color:red;
}
fieldset {
    margin:1px;
    padding:3px;
}
</style>
<script type="text/javascript">
$(function(){
    $("form input").css("border", "2px dotted brown");
});
</script>
</head>
<body>
<form>
    <div>Форма заключена в зеленую рамку</div>
    <label>Ребенок:</label>
    <input type="text" name="name" />
    <fieldset>
        <label>Внук:</label>
        <input type="text" name="newsletter" />
    </fieldset>
</form>
Сестринский элемент по отношению к форме: <input type="text" name="none" />
</body>
</html>
```

## Обсуждение

В HTML-коде, приведенном в листинге 1.2.1, присутствуют три элемента `input`. Наша задача — отыскать только те из них, которые являются наследниками элемента `form`. Указав в селекторе выражение `form input`, мы легко находим только нужные нам элементы и отмечаем их коричневой рамкой. Элемент `input`, расположенный вне пределов тега `<form>`, такой рамкой отмечен не будет.

## Проблема

Необходимо отыскать элементы, являющиеся прямыми потомками какого-либо элемента.

## Решение

Используем селектор `parent > child` для решения этой задачи (листинг 1.2.2).

### Листинг 1.2.2. Использование селектора `parent > child`

```
<script type="text/javascript">
$(function() {
    $("form > input").css("border", "2px dotted brown");
});
</script>
```

## Обсуждение

В листинге 1.2.2 приведен только JavaScript-код, т. к. все остальное осталось без изменений. Поскольку необходимо отыскать только те элементы `input`, которые являются прямыми наследниками `form`, мы решаем задачу, указывая в селекторе выражение `form > input`. Коричневой рамкой в итоге будет отмечен только первый элемент `input`.

## Проблема

Необходимо отыскать элементы, следующие непосредственно за известным элементом.

## Решение

Используем селектор `prev + next` для решения этой задачи (листинг 1.2.3).

**Листинг 1.2.3. Использование селектора `prev + next`**

```
<script type="text/javascript">
$(function(){
    $("label + input").css("border", "2px dotted brown");
});
</script>
```

## Обсуждение

В листинге 1.2.3 приведен только JavaScript-код. Мы указали в селекторе выражение `prev + next` и, таким образом, нашли элементы `input`, которые следуют непосредственно за элементами `label`.

## Проблема

Необходимо отыскать все элементы, располагающиеся на одном уровне (сестринские элементы по отношению друг к другу), следующие непосредственно за известным элементом, который располагается на том же уровне.

## Решение

Используем селектор `prev ~ siblings` для решения этой задачи (листинг 1.2.4).

**Листинг 1.2.4. Использование селектора `prev ~ siblings`**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-2-4</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<style type="text/css">
form {
    border:2px green solid;
    padding:2px;
    margin:0;
    background:#efe;
}
div {
    color:red;
}
```

```
fieldset {
  margin:1px;
  padding:3px;
}
</style>
<script type="text/javascript">
$(function(){
  $("label ~ fieldset").css("border", "2px dotted brown");
});
</script>
</head>
<body>
<form>
  <div>Форма заключена в зеленую рамку</div>
  <label>Ребенок:</label>
  <input type="text" name="name" />
  <fieldset>
    <label>Внук:</label>
    <input type="text" name="newsletter" />
  </fieldset>
</form>
Сестринский элемент по отношению к форме: <input type="text" name="none" />
</fieldset>
  <input type="text" name="email" />
</fieldset>
</body>
</html>
```

## Обсуждение

В листинге 1.2.4 мы немного изменили HTML-код, добавив еще один элемент `fieldset`, внутри которого находится элемент `input`. Указываем в селекторе выражение `label ~ fieldset` и видим, что коричневой рамкой отмечен только тот элемент `fieldset`, который находится внутри формы, поскольку только он, в отличие от `fieldset`, находящегося вне `form`, является сестринским элементом по отношению к `label`.

## 1.3. Основные фильтры

### Проблема

Необходимо установить серый цвет фона только для первой строки в таблице.



## Решение

Для решения задачи используем селектор `:first` (листинг 1.3.1).

### Листинг 1.3.1. Использование фильтра `:first`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-3-1</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<style type="text/css">
table {
    width:400px;
}
</style>
<script type="text/javascript">
$(function() {
    $("tr:first").css("background-color", "#ccc");
});
</script>
</head>
<body>
<table>
<tr>
<td>1-1</td><td>1-2</td><td>1-3</td><td>1-4</td>
</tr>
<tr>
<td>2-1</td><td>2-2</td><td>2-3</td><td>2-4</td>
</tr>
<tr>
<td>3-1</td><td>3-2</td><td>3-3</td><td>3-4</td>
</tr>
<tr>
<td>4-1</td><td>4-2</td><td>4-3</td><td>4-4</td>
</tr>
</table>
</body>
</html>
```

## Обсуждение

HTML-код, приведенный в листинге 1.3.1, не представляет собой ничего интересного — обычная таблица. Посмотрим на JavaScript-код. Указав в селекторе выражение `tr:first` мы смогли установить серый цвет фона только для первой строки таблицы.

## Проблема

Необходимо установить серый цвет фона только для последней строки в таблице.

## Решение

Для решения задачи используем селектор `:last` (листинг 1.3.2).

### Листинг 1.3.2. Использование фильтра `:last`

```
<script type="text/javascript">
$(function() {
  $("tr:last").css("background-color", "#ccc");
});
</script>
```

## Обсуждение

Рассмотрим только JavaScript-код из листинга 1.3.2, поскольку HTML-код остался без изменений. На этот раз мы указали в селекторе выражение `tr:last` и установили серый цвет фона уже для последней строки в таблице.

## Проблема

Необходимо установить серый цвет фона только для четных строк в таблице.

## Решение

Для решения задачи воспользуемся селектором `:even` (листинг 1.3.3).

### Листинг 1.3.3. Использование фильтра `:even`

```
<script type="text/javascript">
$(function() {
  $("tr:even").css("background-color", "#ccc");
});
</script>
```

## Обсуждение

Нам стоит только указать выражение `tr:even` в коде, который приведен в листинге 1.3.3, и все четные строки таблицы (отсчет будет идти от нуля) станут серыми.

## Проблема

Необходимо установить серый цвет фона только для нечетных строк в таблице.

## Решение

Для решения задачи используем селектор `:odd` (листинг 1.3.4).

### Листинг 1.3.4. Использование фильтра `:odd`

```
<script type="text/javascript">
$(function() {
  $("tr:odd").css("background-color", "#ccc");
});
</script>
```

## Обсуждение

Так же просто обстоит дело с отысканием всех нечетных строк. Мы найдем их, всего лишь записав выражение `tr:odd` в коде, который приведен в листинге 1.3.4.

## Проблема

Необходимо отыскать шестую по счету ячейку таблицы.

## Решение

Для решения задачи воспользуемся селектором `:eq(index)` (листинг 1.3.5).

### Листинг 1.3.5. Использование фильтра `:eq(index)`

```
<script type="text/javascript">
$(function() {
  $("td:eq(5)").css("background-color", "#ccc");
});
</script>
```

## Обсуждение

В примере из листинга 1.3.5 мы нашли шестую ячейку таблицы, указав в селекторе выражение `td:eq(5)`, потому что отсчет идет, начиная от нуля.

## Проблема

Необходимо отыскать все ячейки таблицы, которые следуют после шестой по счету (иначе говоря — с индексом, более пяти).

## Решение

Для решения задачи используем селектор `:gt(index)` (листинг 1.3.6).

### Листинг 1.3.6. Использование фильтра `:gt(index)`

```
<script type="text/javascript">
$(function() {
  $("td:gt(5)").css("background-color", "#ccc");
});
</script>
```

## Обсуждение

Все очень просто, как код в листинге 1.3.6. Указываем в селекторе выражение `td:gt(5)`, и все ячейки с индексом более пяти стали серого цвета.

## Проблема

Необходимо отыскать все ячейки таблицы, которые следуют перед шестой по счету (иначе говоря — с индексом, менее пяти).

## Решение

Для решения задачи применим селектор `:lt(index)` (листинг 1.3.7).

### Листинг 1.3.7. Использование фильтра `:lt(index)`

```
<script type="text/javascript">
$(function() {
  $("td:lt(5)").css("background-color", "#ccc");
});
</script>
```

## Обсуждение

Тоже проще простого. Указываем в селекторе выражение `td:lt(5)`, и все ячейки перед шестой стали серыми.

## Проблема

Необходимо отыскать на веб-странице все элементы, являющиеся заголовками, например, `h1`, `h2`, `h3` и т. д.

## Решение

Для решения задачи используем селектор `:header` (листинг 1.3.8).

### Листинг 1.3.8. Использование фильтра `:header`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-3-8</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<style type="text/css">
table {
    width:400px;
}
</style>
<script type="text/javascript">
$(function(){
    $(".header").css("background-color","#ccc");
});
</script>
</head>
<body>
<h1>Заголовок h1</h1>
<p>Некоторый текст в элементе p</p>
<h3>Заголовок h3</h3>
<ul>
<li>Список: пункт 1</li>
<li>Список: пункт 2</li>
<li>Список: пункт 3</li>
</ul>
</body>
</html>
```

```
</ul>
</body>
</html>
```

## Обсуждение

В HTML-коде, который приведен в листинге 1.3.8, присутствуют элемент параграф `p`, список `ul` и два элемента, являющиеся заголовками, — `h1` и `h3`. Указывая в селекторе выражение `:header`, мы имеем возможность легко отыскать эти заголовки и установить для них серый цвет фона.

## 1.4. Фильтрация по содержимому

### Проблема

Необходимо отыскать все элементы, внутри которых находится текст, содержащий подстроку 'John'.

### Решение

Для решения задачи используем фильтр `:contains` (листинг 1.4.1).

#### Листинг 1.4.1. Использование фильтра `:contains`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">
<head>
<title>example-1-4-1</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="../../js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    $("div:contains('John')").css("text-decoration", "underline");
});
</script>
</head>
<body>
<div>John Resig</div>
<div>George Martin</div>
<div>Malcom John Sinclair</div>
<div>J. Ohn</div>
```