

ГЕННАДИЙ САМКОВ



jQuery

Сборник рецептов

2-е издание



МЕТОДЫ
И ВСПОМОГАТЕЛЬНЫЕ
ФУНКЦИИ БИБЛИОТЕКИ jQuery

ВЗАИМОДЕЙСТВИЕ jQuery
И AJAX

НАДСТРОЙКА UI jQuery

САМЫЕ ПОПУЛЯРНЫЕ
ПЛАГИНЫ ДЛЯ jQuery

PRO

ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ



Геннадий Самков

jQuery

Сборник рецептов

2-е издание

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.068+800.92 jQuery
ББК 32.973.26-018.1
С17

Самков Г. А.

С17 jQuery. Сборник рецептов. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 416 с.: ил. + CD-ROM — (Профессиональное программирование)

ISBN 978-5-9775-0732-5

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 надстройке UI jQuery. Приведено большое количество примеров использования плагинов для jQuery — создание графиков и диаграмм, фотогалерей, меню, работа с таймерами и cookies, обработка табличных данных и др. Во втором издании в примерах используется библиотека jQuery версий 1.4.4 и 1.5.2, а также надстройка UI jQuery 1.8.9. Компакт-диск содержит примеры из книги, файлы библиотеки jQuery 1.4.4 и 1.5.2, файлы надстройки UI jQuery 1.8.9, а также файлы 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 30.06.11.

Формат 70×100^{1/16}. 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 978-5-9775-0732-5

© Самков Г. А., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

Введение	7
Структура книги.....	7
Как работать с книгой.....	8
Источники информации	9
Благодарности	10
ЧАСТЬ I. МЕТОДЫ БИБЛИОТЕКИ JQUERY	11
Глава 1. Выбор элементов	13
1.1. Базовые правила.....	13
1.2. Выбор элементов с учетом иерархии	20
1.3. Основные фильтры	24
1.4. Фильтрация по содержимому	30
1.5. Фильтры видимых и невидимых элементов	34
1.6. Селекторы атрибутов.....	36
1.7. Фильтры элементов форм	40
1.8. Фильтры состояния элементов форм	43
1.9. Фильтры элементов-потомков	47
Глава 2. Атрибуты элементов	53
2.1. Управление атрибутами элементов	53
2.2. Работа с атрибутом <i>class</i>	57
2.3. Работа с HTML и текстом	59
2.4. Работа с атрибутом <i>value</i>	61
Глава 3. Визуальные эффекты	67
3.1. Как показывать и скрывать элементы.....	67
3.2. Эффекты "скольжения" и "затухания"	70
3.3. Создание анимации.....	74
3.4. Эффекты UI jQuery	79
Глава 4. Работа с CSS-свойствами	84
4.1. Как получать и устанавливать значения CSS-свойств элементов.....	84
4.2. Ширина и высота элементов	88
4.3. Позиционирование элементов	90

Глава 5. Работа с данными в jQuery	94
5.1. Сохранение и извлечение данных	94
Глава 6. Манипуляции над элементами	98
6.1. Изменение содержимого элементов	98
6.2. Как вставлять элементы в DOM	101
6.3. Замена, удаление и копирование элементов	111
Глава 7. Перемещение по элементам	118
7.1. Поиск нужных элементов в DOM.....	118
7.2. Фильтрация элементов набора.....	128
7.3. Прочие методы.....	138
Глава 8. События и их обработка	144
8.1. События документа.....	144
8.2. Назначение, удаление и вызов событий.....	146
8.3. События мыши, клавиатуры, браузера и форм	156
Глава 9. Взаимодействие jQuery и AJAX	165
9.1. Сокращенные методы.....	165
9.2. Вспомогательные функции <i>\$.ajax()</i> и <i>\$.ajaxSetup()</i>	178
9.3. События AJAX	187
9.4. Полезные вспомогательные функции	192
Глава 10. Полезные вспомогательные функции и методы jQuery	195
10.1. Некоторые операции с массивами и объектами в jQuery	195
10.2. Некоторые операции с наборами элементов jQuery	208
10.3. Другие полезные вспомогательные функции	214
ЧАСТЬ II. РАСШИРЕНИЯ ДЛЯ БИБЛИОТЕКИ JQUERY	219
Глава 11. Меню для веб-сайта	221
11.1. Плагин jQuery Superfish.....	221
Глава 12. Работа с таблицами	230
12.1. Плагин jqGrid	230
Глава 13. Графики и диаграммы	250
13.1. Плагин jqPlot	250
Глава 14. AJAX-формы	262
14.1. Плагин jQuery Form	262
14.2. Плагин jQuery Validate	267
14.3. Плагин jQuery Uploadify.....	273
Глава 15. Фотогалерея для сайта	282
15.1. Фотогалерея FancyBox	282
Глава 16. Несколько полезных плагинов	297
16.1. jQuery Cookie.....	297

16.2. jQuery Timers	299
16.3. jQuery Cluetip.....	302
Глава 17. UI jQuery — виджеты.....	308
17.1. Виджет Accordion	308
17.2. Виджет DatePicker.....	318
17.3. Виджет Dialog	329
17.4. Виджет Progressbar	337
17.5. Виджет Slider.....	340
17.6. Виджет Tabs	346
17.7. Виджет Button	356
17.8. Виджет Autocomplete.....	361
Глава 18. UI jQuery — взаимодействие с элементами страницы.....	370
18.1. Draggable — перемещение элементов.....	370
18.2. Droppable — "сброс" элементов	379
18.3. Resizable — изменение размеров элементов	386
18.4. Selectable — выбор элементов	392
18.5. Sortable — сортировка элементов	399
Приложение. Описание компакт-диска	409
Литература	411
Предметный указатель	412

Введение

Предлагаемая книга представляет собой сборник примеров, поясняющих возможности большинства методов, предоставляемых API популярной JavaScript-библиотеки jQuery. Много внимания уделено UI jQuery — надстройке над библиотекой, применяемой при проектировании пользовательских интерфейсов. Также в книгу включены подробные описания и рекомендации по использованию наиболее востребованных плагинов, которые пригодятся при решении задач, часто встречающихся в программировании.

Предполагается, что читатель знаком с CSS, HTML и основами JavaScript. Кроме того, потребуются некоторые начальные знания 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 элементов — базовые правила, иерархия, фильтры, селекторы атрибутов, фильтры элементов форм;
- работа с атрибутами и содержимым элементов, работа с данными в элементах форм;
- создание визуальных эффектов, в том числе с помощью надстройки UI jQuery;
- работа с CSS-свойствами элементов;
- работа с данными в jQuery, в том числе с использованием HTML 5;

- ❑ манипуляции элементами — изменение содержимого, вставка, замена, удаление и копирование элементов DOM;
- ❑ перемещение по элементам DOM — поиск и фильтрация элементов, управление цепочками вызовов функций;
- ❑ работа с событиями — назначение, удаление и вызов событий для мыши, клавиатуры, браузера и форм;
- ❑ взаимодействие jQuery и AJAX;
- ❑ некоторые полезные вспомогательные функции и методы 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:

- ❑ вертикальное и горизонтальное многоуровневое меню на основе плагина jQuery Superfish;
- ❑ организация работы с данными, представленными в табличной форме, рассматривается на примере плагина jqGrid;
- ❑ возможности реализации графиков и диаграмм на страницах веб-сайта демонстрируются на примере плагина jqPlot;
- ❑ работа с AJAX-формами — плагины jQuery Form (построение AJAX-формы), jQuery Validate (проверка данных в AJAX-форме) и FileUpload (загрузка файлов на сервер);
- ❑ фотогалереи для веб-сайтов рассматриваются на примере плагина FancyBox;
- ❑ некоторые полезные плагины — 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), Tabs (организация переключения вкладок), Button (стилизиция и управление поведением кнопок и некоторых элементов форм), Autocomplete (список подсказок);
- ❑ 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, работающего 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

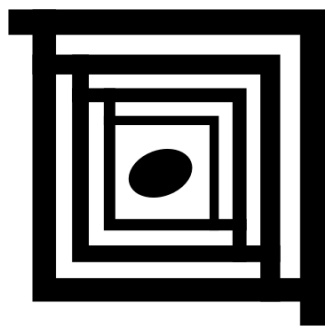
В книге невозможно охватить все вопросы, и читателю наверняка потребуются дополнительные сведения, например, из Интернета. Кроме того, могут изменяться версии программного обеспечения, рассматриваемого в книге. Вот адреса, которыми вы можете воспользоваться:

- ❑ <http://jquery.com/> — официальный сайт библиотеки jQuery (англ.);
- ❑ <http://api.jquery.com/> — оригинальная документация по библиотеке jQuery (англ.);
- ❑ <http://jqueryui.com/> — официальный сайт UI jQuery (англ.);
- ❑ <http://www.linkexchanger.su> — блог автора книги, содержит много статей с примерами использования библиотеки jQuery (рус.);
- ❑ <http://www.linkexchanger.su/forum/> — форум, посвященный вопросам разработки с применением библиотеки jQuery, UI jQuery и плагинов для библиотеки (рус.);
- ❑ <http://slyweb.ru/jquerymain/> — перевод документации jQuery (рус.);
- ❑ <http://plugins.jquery.com/> — на сайте представлено большое количество разнообразных плагинов для библиотеки jQuery (англ.);
- ❑ http://users.tpg.com.au/j_birch/plugins/superfish/ — страница плагина jQuery Superfish (англ.);
- ❑ <http://trirand.com/blog/jqgrid/jqgrid.html> — демо-галерея плагина jqGrid (англ.);
- ❑ <http://www.trirand.com/jqgridwiki/doku.php?id=wiki:jqgriddocs> — официальная документация на плагин jqGrid (англ.);
- ❑ <http://www.jqplot.com> — сайт плагина jqPlot (англ.);
- ❑ <http://malsup.com/jquery/form/> — страница 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.offput.ca/timers/> — страница плагина jQuery Timers (англ.);
- ❑ <http://firebug.ru> — сайт, посвященный Firebug, замечательному средству отладки JavaScript-кода (рус.).

Благодарности

Автор приносит искренние благодарности Владимиру Кондратьеву за его готовность поделиться своим богатым практическим опытом работы с плагинами, Игорю Пирогову и Андрею Зайцеву — за их поддержку и квалифицированные технические консультации, Эдуарду Аметову — за консультации по вопросам работы с изображениями.

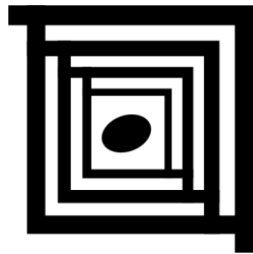


ЧАСТЬ I

Методы библиотеки jQuery

Глава 1.	Выбор элементов
Глава 2.	Атрибуты элементов
Глава 3.	Визуальные эффекты
Глава 4.	Работа с CSS-свойствами
Глава 5.	Работа с данными в jQuery
Глава 6.	Манипуляции над элементами
Глава 7.	Перемещение по элементам
Глава 8.	События и их обработка
Глава 9.	Взаимодействие jQuery и AJAX
Глава 10.	Полезные вспомогательные функции и методы 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 выбрано, мы будем рассматривать это как набор элементов, называя его объектом jQuery.

1.1. Базовые правила

ЗАДАЧА

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-1.5.2.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 объекте 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.

ЗАДАЧА

Необходимо отыскать все элементы веб-страницы, но только в контексте `body`, исключив, таким образом, все прочие элементы.

Решение

Для решения этой задачи также 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-1.5.2.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 13 в примере из листинга 1.1.1. В набор попали только элементы, находящиеся внутри `body`. Сам элемент `body` в набор не попал.

ЗАДАЧА

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-1.5.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
  $("#myDiv").css("border", "1px solid #f00");
});
</script>
<style type="text/css">
div {
  float:left;
  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.

ПРИМЕЧАНИЕ

Не забывайте важный момент — в пределах документа значение атрибута `id` обязательно должно быть уникальным.

ЗАДАЧА

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-1.5.2.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 {
    float:left;
    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. Элементу с идентификатором `my.Div` мы установили красную рамку, а элементу с идентификатором `my[Div]` — зеленую.

ЗАДАЧА

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

Решение

Для решения задачи достаточно 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-1.5.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
```

```

$(function() {
    $("div").css("border", "2px dotted #f00");
});
</script>
<style type="text/css">
div, 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

Обратите внимание — с помощью нескольких CSS-правил мы задали совершенно одинаковое оформление как для элементов `p`, так и для `div`. Но мы добавили и JavaScript-код и с помощью селектора `element` нам удалось выбрать все элементы `div`, имеющиеся на веб-странице. Используя метод `css()`, мы установили для выбранных элементов рамки из точек красного цвета шириной 2 пх.

ЗАДАЧА

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-1.5.2.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 атрибута `class`. Для наглядности вновь применяем метод `css()`, чтобы установить для выбранных элементов рамки красного цвета шириной 1 px.

ЗАДАЧА

Используя только один селектор, необходимо выбрать разные элементы по имени класса, значению идентификатора и названию элемента.

Решение

Для решения задачи применяется составной селектор (листинг 1.1.7).

Листинг 1.1.7. Использование составного 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-1-1-7</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="js/jquery-1.5.2.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
  $("#item2, p.test, div").css("background-color", "#ccc");
});
</script>
<style type="text/css">
```

```

div,p,li { border:1px solid #00f; margin:2px; }
</style>
</head>
<body>
<div>Элемент div</div>
<p>Элемент p</p>
<div>Элемент div</div>
<p class="test">Элемент p class="test"</p>
<p>Элемент p</p>
<ul id="test">
  <li id="item1">Первый пункт списка id="item1"</li>
  <li id="item2">Второй пункт списка id="item2"</li>
</ul>
<p class="test">Элемент p class="test"</p>
<div>Элемент div</div>
</body>
</html>

```

Обсуждение

С помощью составного селектора мы смогли сразу выбрать все элементы `div`, элементы `p` с классом `test` и элемент `li` по его идентификатору `item2`. Для всех элементов, попавших в этот набор, устанавливаем серый цвет фона методом `css()`.

1.2. Выбор элементов с учетом иерархии

ЗАДАЧА

Необходимо отыскать элементы, являющиеся потомками какого-либо элемента.

Решение

Селектор `ancestor descendant` поможет решить эту задачу (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-1.5.2.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 выделен не будет.

ЗАДАЧА

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 элементы `input`, являющиеся прямыми наследниками `form`, мы указываем в селекторе выражение `form > input`. Коричневой рамкой в итоге будет отмечен только первый элемент `input`.

ЗАДАЧА

Необходимо отыскать элементы, следующие непосредственно за известным элементом.

Решение

Решить эту задачу позволит селектор `prev + next` (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`. Коричневая рамка досталась лишь двум элементам `input` внутри формы, поскольку только перед ними есть элемент `label`. Элемент `input`, находящийся вне тега `form`, не имеет перед собой элемента `label`, поэтому и не был выбран.

ЗАДАЧА

Необходимо отыскать все элементы, располагающиеся на одном уровне с некоторым известным элементом, иными словами — сестринские элементы.

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-1.5.2.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

ЗАДАЧА

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

Решение

Решить задачу поможет фильтр `: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-1.5.2.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.

ЗАДАЧА

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

Решение

Для решения задачи применим фильтр `: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.

ЗАДАЧА

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

Решение

Для решения задачи воспользуемся фильтром `: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.

ЗАДАЧА

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

Решение

Для решения задачи используем фильтр `: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.

ЗАДАЧА

Необходимо отыскать шестую по счету ячейку таблицы.

Решение

Для решения задачи воспользуемся фильтром `: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.

ЗАДАЧА

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

Решение

Решить задачу поможет фильтр `: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.

ЗАДАЧА

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

Решение

Для решения задачи применим фильтр `: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.

ЗАДАЧА

Необходимо отыскать все ячейки таблицы, кроме одной, индекс которой равен пяти.

Решение

Для решения задачи применим фильтр `:not()` (листинг 1.3.8).

Листинг 1.3.8. Использование фильтра `:not()`

```
<script type="text/javascript">
$(function() {
  $("td:not(:eq(5))").css("background-color", "#ccc");
});
</script>
```

Обсуждение

Внутри фильтра `:not()` мы указали другой фильтр — `:eq(5)`, получив таким образом выражение "все, кроме". В результате все ячейки таблицы кроме одной стали серыми.

ЗАДАЧА

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

Решение

Решить задачу позволяет фильтр `:header` (листинг 1.3.9).

Листинг 1.3.9. 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-1.5.2.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>
```

Обсуждение

В HTML-коде, который приведен в листинге 1.3.9, присутствуют элемент параграфа `p`, список `ul` и два элемента, являющиеся заголовками, — `h1` и `h3`. Указывая в селекторе фильтр `:header`, мы имеем возможность легко отыскать эти заголовки и установить для них серый цвет фона.

ЗАДАЧА

Необходимо выбрать все элементы, которые к моменту выбора будут находиться в процессе анимации.

Решение

Для решения задачи применим фильтр `:animated()` (листинг 1.3.10).