

ГЕННАДИЙ САМКОВ



jQuery

Сборник рецептов

МЕТОДЫ
И ВСПОМОГАТЕЛЬНЫЕ
ФУНКЦИИ БИБЛИОТЕКИ jQuery

ВЗАИМОДЕЙСТВИЕ
jQuery И AJAX

НАДСТРОЙКА UI jQuery

САМЫЕ ПОПУЛЯРНЫЕ
ПЛАГИНЫ ДЛЯ jQuery



PRO

ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ

УДК 681.3.068+800.92 jQuery

ББК 32.973.26-018.1

C17

Самков Г. А.

C17 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 надстройке 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 надстройки 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^{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 978-5-9775-0495-9

© Самков Г. А., 2009

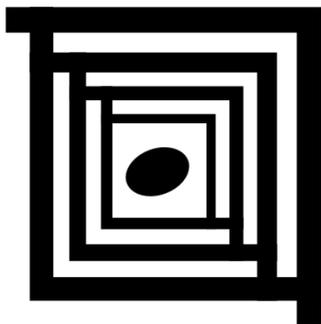
© Оформление, издательство "БХВ-Петербург", 2009

Оглавление

Введение	1
Структура книги.....	1
Как работать с книгой.....	2
Источники информации.....	3
Благодарности.....	4
ЧАСТЬ I. МЕТОДЫ БИБЛИОТЕКИ JQUERY	5
Глава 1. Выбор элементов	7
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
Глава 2. Атрибуты элементов	48
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
Глава 3. Визуальные эффекты	64
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

Глава 4. Работа с CSS-свойствами	81
4.1. Как получать и устанавливать значения CSS-свойств элементов.....	81
4.2. Ширина и высота элементов.....	85
4.3. Позиционирование элементов.....	88
Глава 5. Некоторые методы ядра библиотеки jQuery	92
5.1. Примеры работы с объектом jQuery.....	92
5.2. Сохранение и извлечение данных.....	97
Глава 6. Манипуляции над элементами	101
6.1. Изменение содержимого элементов.....	101
6.2. Как вставлять элементы в DOM.....	103
6.3. Замена, удаление и копирование элементов.....	113
Глава 7. Перемещение по элементам	120
7.1. Поиск нужных элементов в DOM.....	120
7.2. Фильтрация элементов набора.....	131
7.3. Перемещение по цепочке вызовов.....	138
Глава 8. События и их обработка	142
8.1. Готовность документа.....	142
8.2. Назначение, удаление и вызов событий.....	144
8.3. Взаимодействие с элементами.....	152
8.4. События.....	155
Глава 9. Взаимодействие jQuery и AJAX	161
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
Глава 10. Утилиты jQuery	190
10.1. Некоторые операции с массивами и объектами в jQuery.....	190
ЧАСТЬ II. РАСШИРЕНИЯ ДЛЯ БИБЛИОТЕКИ JQUERY	201
Глава 11. Меню для веб-сайта	203
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

Глава 12. Работа с таблицами	219
12.1. Плагин jQuery DataTables.....	219
Глава 13. Графики и диаграммы	229
13.1. Плагин jqPlot.....	229
Глава 14. AJAX-формы	242
14.1. Плагин jQuery Autocomplete.....	242
14.2. Плагин jQuery Form.....	252
14.3. Плагин jQuery Validate.....	257
14.4. Плагин jQuery Uploadify.....	264
Глава 15. Фотогалереи для сайта	274
15.1. Фотогалерея FancyBox.....	274
15.2. Простая фотогалерея.....	282
Глава 16. Несколько полезных плагинов	286
16.1. jQuery Cookie.....	286
16.2. jQuery Corner.....	288
16.3. jQuery Timers.....	293
16.4. jQuery Cluetip.....	297
Глава 17. UI jQuery — виджеты	303
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
Глава 18. UI jQuery — взаимодействие с элементами страницы	354
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
Литература	397
Приложение. Описание компакт-диска	398
Предметный указатель	400

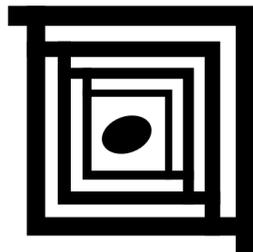


ЧАСТЬ I

Методы библиотеки jQuery

Глава 1.	Выбор элементов
Глава 2.	Атрибуты элементов
Глава 3.	Визуальные эффекты
Глава 4.	Работа с CSS-свойствами
Глава 5.	Некоторые методы ядра библиотеки 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.	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>
```