

# Содержание

<b>Предисловие</b> .....	14
<b>Вступление</b> .....	16
<b>Благодарности</b> .....	18
<b>Об этой книге</b> .....	19
<b>Об иллюстрации на обложке</b> .....	24
<b>Часть I. Простые расширения</b> .....	25
<b>Глава 1. Расширения для jQuery</b> .....	26
1.1. История развития jQuery.....	26
1.1.1. Происхождение.....	27
1.1.2. Развитие.....	28
1.1.3. Современное состояние дел.....	31
1.2. Расширение jQuery.....	32
1.2.1. Что доступно для расширения?.....	32
1.3. Примеры расширений.....	36
1.3.1. jQuery UI.....	37
1.3.2. Проверка данных.....	38
1.3.3. Графический ползунок.....	39
1.3.4. Интеграция со службой Google Map.....	40
1.3.5. Cookies.....	41
1.3.6. Анимация, основанная на изменении цвета.....	42
1.4. В заключение.....	43
<b>Глава 2. Первое расширение</b> .....	44
2.1. Архитектура jQuery.....	44
2.1.1. Точки интеграции с библиотекой jQuery.....	46
2.1.2. Селекторы.....	47
2.1.3. Расширения коллекций.....	48
2.1.4. Вспомогательные функции.....	49
2.1.5. Виджеты jQuery UI.....	49
2.1.6. Эффекты jQuery UI.....	50
2.1.7. Анимация свойств.....	50
2.1.8. Поддержка Ajax.....	51
2.1.9. Обработка событий.....	52

2.1.10. Правила проверки данных .....	52
2.2. Простое расширение .....	53
2.2.1. Текст подсказки .....	53
2.2.2. Реализация расширения Watermark .....	54
2.2.3. Удаление текста подсказок .....	56
2.2.4. Применение расширения Watermark .....	57
2.3. В заключение .....	60
<b>Глава 3. Селекторы и фильтры .....</b>	<b>61</b>
3.1. Что такое селекторы и фильтры?.....	62
3.1.1. Зачем добавлять новые селекторы?.....	62
3.1.2. Простые селекторы .....	63
3.1.3. Селекторы псевдоклассов.....	65
3.2. Добавление нового селектора псевдокласса.....	70
3.2.1. Структура селектора псевдокласса .....	70
3.2.2. Добавление селектора по точному соответствию содержимого .....	72
3.2.3. Добавление селектора по соответствию шаблону.....	75
3.2.4. Добавление селектора по типу элемента.....	77
3.2.5. Добавление селектора элементов с текстом на иностранном языке.....	78
3.2.6. Селекторы из расширения Validation.....	80
3.3. Добавление фильтров множеств .....	81
3.3.1. Структура селектора множества .....	81
3.3.2. Добавление селектора выборки элементов из середины множества.....	83
3.3.3. Расширение селектора по индексу .....	84
3.4. В заключение .....	87
<b>Часть II. Расширения и функции .....</b>	<b>88</b>
<b>Глава 4. Принципы разработки расширений .....</b>	<b>89</b>
4.1. Архитектура расширений .....	89
4.1.1. Преимущества оформления расширений в виде модулей .....	90
4.1.2. Проектирование архитектуры.....	91
4.1.3. Поддержка модульной архитектуры в расширениях.....	92
4.2. Руководящие принципы .....	93
4.2.1. Нарращивайте возможности прогрессивно.....	93

4.2.2. Объявляйте только одно имя и используйте его повсюду .....	93
4.2.3. Помещайте все в объект jQuery .....	95
4.2.4. Не рассчитывайте, что имя \$ будет ссылаться на jQuery .....	96
4.2.5. Скрывайте тонкости реализации с использованием областей видимости .....	96
4.2.6. Используйте методы для доступа к дополнительной функциональности .....	98
4.2.7. Возвращайте объект jQuery, если это возможно .....	98
4.2.8. Используйте функцию data для сохранения данных экземпляра .....	99
4.2.9. Предусматривайте возможность настройки .....	100
4.2.10. Используйте осмысленные значения по умолчанию .....	101
4.2.11. Добавьте поддержку локализации .....	103
4.2.12. Реализуйте оформление внешнего вида с помощью CSS .....	104
4.2.13. Тестируйте расширение в основных браузерах .....	107
4.2.14. Создавайте комплекты повторимых тестов .....	108
4.2.15. Создавайте демонстрационные примеры и документацию .....	108
4.3. В заключение .....	110
<b>Глава 5. Расширения коллекций</b> .....	<b>112</b>
5.1. Что такое расширения коллекций? .....	112
5.2. Инфраструктура поддержки расширений .....	113
5.2.1. Расширение MaxLength .....	113
5.2.2. Устройство расширения MaxLength .....	114
5.3. Определение собственного расширения .....	117
5.3.1. Выбор имени .....	117
5.3.2. Инкапсуляция .....	118
5.3.3. Использование объекта-одиночки .....	119
5.4. Применение к элементам .....	121
5.4.1. Простое подключение .....	121
5.4.2. Инициализация расширения .....	122
5.4.3. Вызов методов .....	124
5.4.4. Методы чтения .....	126
5.5. Параметры настройки .....	128

5.5.1. Значения настроек по умолчанию.....	129
5.5.2. Локализация.....	130
5.5.3. Реакция на изменение параметров.....	132
5.5.4. Реализация реакции на изменение параметров в MaxLength .....	135
5.5.5. Активация и деактивация виджета .....	136
5.6. Добавление обработчиков событий.....	138
5.6.1. Регистрация обработчиков событий.....	138
5.6.2. Вызов обработчика события.....	139
5.7. Добавление методов .....	140
5.7.1. Получение текущей длины .....	140
5.8. Удаление расширения.....	141
5.8.1. Метод destroy .....	141
5.9. Заключительные штрихи .....	142
5.9.1. Главная цель расширения.....	142
5.9.2. Реализация поддержки стилей .....	145
5.10. Законченное расширение.....	146
5.11. В заключение.....	148
<b>Глава 6. Расширения-функции.....</b>	<b>149</b>
6.1. Определение расширения .....	150
6.1.1. Расширение для локализации.....	150
6.1.2. Код инфраструктуры.....	152
6.1.3. Загрузка локализаций.....	152
6.2. Расширение Cookie.....	156
6.2.1. Операции с данными cookie.....	157
6.2.2. Чтение и запись cookies.....	158
6.3. В заключение .....	161
<b>Глава 7. Тестирование, упаковка и документирование расширений .....</b>	<b>162</b>
7.1. Тестирование расширений.....	163
7.1.1. Что тестировать?.....	164
7.1.2. Использование QUnit.....	165
7.1.3. Тестирование расширения MaxLength .....	167
7.1.4. Тестирование установки и извлечения параметров расширения.....	170
7.1.5. Имитация действий пользователя.....	172
7.1.6. Тестирование функций-обработчиков.....	174
7.2. Упаковка расширений.....	176

7.2.1. Сборка всех файлов вместе .....	176
7.2.2. Минификация расширения .....	178
7.2.3. Реализация простого примера .....	181
7.3. Документирование расширений .....	184
7.3.1. Документирование параметров настройки .....	184
7.3.2. Документирование методов и вспомогательных функций .....	185
7.3.3. Демонстрация возможностей расширения.....	186
7.4. В заключение .....	188
<b>Часть III. Расширение jQuery UI .....</b>	<b>190</b>
<b>Глава 8. Виджеты jQuery UI .....</b>	<b>191</b>
8.1. Инфраструктура поддержки виджетов .....	192
8.1.1. Модули jQuery UI .....	192
8.1.2. Модуль Widget .....	194
8.1.3. Расширение MaxLength .....	196
8.1.4. Устройство расширения MaxLength .....	197
8.2. Определение виджета .....	198
8.2.1. Выбор имени .....	199
8.2.2. Инкапсуляция виджета .....	199
8.2.3. Объявление виджета .....	200
8.3. Применение к элементам.....	202
8.3.1. Простое подключение и инициализация .....	202
8.4. Параметры настройки.....	204
8.4.1. Значения настроек по умолчанию .....	205
8.4.2. Реакция на изменение параметров .....	206
8.4.3. Реализация параметров настройки MaxLength .....	208
8.4.4. Активация и деактивация виджета .....	212
8.5. Добавление обработчиков событий .....	213
8.5.1. Регистрация обработчиков событий.....	213
8.5.2. Вызов обработчиков событий .....	215
8.6. Добавление методов .....	216
8.6.1. Получение текущей длины .....	217
8.7. Удаление виджета.....	218
8.7.1. Метод <code>_destroy</code> .....	218
8.8. Заключительные штрихи .....	220
8.8.1. Главная цель расширения.....	220
8.8.2. Реализация поддержки стилей .....	222
8.9. Законченное расширение .....	224
8.10. В заключение .....	226

<b>Глава 9. Взаимодействия с мышью в jQuery UI</b> .....	228
9.1. Модуль jQuery UI Mouse .....	229
9.1.1. Операции буксировки мышью .....	229
9.1.2. Параметры настройки, поддерживаемые модулем Mouse .....	229
9.2. Определение виджета .....	231
9.2.1. Функциональность виджета Signature .....	231
9.2.2. Устройство расширения Signature .....	233
9.2.3. Объявление виджета .....	235
9.3. Применение расширения к элементу .....	236
9.3.1. Инициализация, выполняемая инфраструктурой .....	236
9.3.2. Собственная инициализация .....	237
9.4. Параметры настройки .....	239
9.4.1. Значения настроек по умолчанию .....	240
9.4.2. Установка параметров .....	241
9.4.3. Реализация параметров настройки Signature .....	242
9.4.4. Активация и деактивация виджета .....	244
9.5. Добавление обработчиков событий .....	244
9.5.1. Регистрация обработчиков событий .....	245
9.5.2. Вызов обработчиков событий .....	246
9.6. Взаимодействие с мышью .....	246
9.6.1. Можно ли начать буксировку? .....	247
9.6.2. Начало буксировки .....	248
9.6.3. Слежение за положением указателя в процессе буксировки .....	249
9.6.4. Завершение буксировки .....	250
9.7. Добавление методов .....	250
9.7.1. Очистка подписи .....	251
9.7.2. Преобразование в формат JSON .....	252
9.7.3. Повторное отображение подписи .....	253
9.7.4. Проверка наличия подписи .....	254
9.8. Удаление виджета .....	255
9.8.1. Метод <code>_destroy</code> .....	256
9.9. Законченное расширение .....	257
9.10. В заключение .....	258
<b>Глава 10. Эффекты jQuery UI</b> .....	259
10.1. Инфраструктура поддержки эффектов в jQuery UI .....	260
10.1.1. Модуль Effects .....	260
10.1.2. Общие функции эффектов .....	262

10.1.3. Существующие эффекты.....	265
10.2. Добавление нового эффекта.....	267
10.2.1. Эффект сжатия.....	267
10.2.2. Инициализация эффекта.....	269
10.2.3. Реализация эффекта.....	271
10.2.4. Реализация эффекта в версиях jQuery UI ниже 1.9.....	273
10.2.5. Законченный эффект.....	274
10.3. Функции управления переходами.....	275
10.3.1. Что такое «функция управления переходом»?.....	275
10.3.2. Существующие функции управления переходами.....	277
10.3.3. Добавление новой функции управления переходом.....	279
10.4. В заключение.....	282
<b>Часть IV. Прочие расширения.....</b>	<b>284</b>
<b>Глава 11. Анимация свойств.....</b>	<b>285</b>
11.1. Инфраструктура поддержки анимационных эффектов.....	286
11.1.1. Механизм анимации.....	287
11.1.2. Порядок выполнения анимации.....	289
11.2. Добавление собственного обработчика анимации свойства.....	292
11.2.1. Анимация свойства background-position.....	293
11.2.2. Объявление обработчика и извлечение значения свойства.....	294
11.2.3. Изменение значения свойства.....	297
11.2.4. Анимация свойства background-position в jQuery 1.7....	299
11.2.5. Законченное расширение.....	301
11.3. В заключение.....	301
<b>Глава 12. Расширение поддержки Ajax.....</b>	<b>303</b>
12.1. Инфраструктура поддержки Ajax.....	304
12.1.1. Предварительные фильтры.....	305
12.1.2. Транспорт.....	306
12.1.3. Преобразователи.....	307
12.2. Добавление предварительного фильтра.....	308
12.2.1. Изменение типа данных.....	308
12.2.2. Отмена запроса Ajax.....	309
12.3. Добавление транспорта Ajax.....	310
12.3.1. Загрузка изображений.....	311
12.3.2. Имитация загрузки HTML для нужд тестирования.....	314

12.4. Добавление преобразователя Ajax .....	318
12.4.1. Формат CSV .....	318
12.4.2. Преобразование текста в формат CSV .....	319
12.4.3. Преобразование данных CSV в таблицу .....	324
12.5. Расширения Ajax .....	325
12.6. В заключение .....	326
<b>Глава 13. Расширение поддержки событий</b> .....	<b>328</b>
13.1. Инфраструктура поддержки специализированных событий .....	329
13.1.1. Подключение обработчиков событий .....	330
13.1.2. Возбуждение событий .....	331
13.2. Добавление специализированного события .....	332
13.2.1. Добавление события щелчка правой кнопкой мыши .....	333
13.2.2. Запрет передачи события щелчка правой кнопкой мыши .....	336
13.2.3. Событие многократных щелчков правой кнопкой .....	337
13.2.4. Функции для взаимодействия с событиями .....	342
13.3. Расширение существующих событий .....	343
13.3.1. Добавление поддержки правой кнопки в событие click .....	344
13.4. Другие функциональные возможности событий .....	346
13.4.1. Реакция по умолчанию на события .....	346
13.4.2. Функции обратного вызова preDispatch и postDispatch .....	347
13.4.3. Предотвращение всплытия события .....	348
13.4.4. Автоматическое связывание и делегирование .....	349
13.5. В заключение .....	351
<b>Глава 14. Создание правил проверки</b> .....	<b>352</b>
14.1. Расширение Validation .....	353
14.1.1. Назначение правил проверки .....	354
14.2. Добавление новых правил проверки .....	356
14.2.1. Добавление правила проверки соответствия шаблону .....	357
14.2.2. Генерирование правил сопоставления с шаблоном .....	360
14.3. Добавление правила для проверки нескольких полей .....	363
14.3.1. Группировка полей .....	363
14.3.2. Определение правила для группы полей .....	364
14.4. В заключение .....	367



---

<b>Приложение А. Регулярные выражения</b> .....	369
A.1. Основы регулярных выражений.....	370
A.2. Синтаксис регулярных выражений .....	371
A.3. Функции объекта RegExp.....	375
A.4. Функции объекта String .....	376
A.5. Приемы применения .....	377
A.5.1. Проверка данных .....	377
A.5.2. Извлечение информации .....	378
A.5.3. Обработка нескольких совпадений.....	378
A.6. В заключение.....	379
<b>Глоссарий</b> .....	380
<b>Алфавитный указатель</b> .....	389

# Предисловие

С момента появления в 2006 году jQuery превратилась в самую популярную библиотеку JavaScript, используемую для управления документами HTML и наполнения их динамическими особенностями. Врожденная поддержка самых разных браузеров в jQuery позволяет разработчикам сосредоточиться на создании веб-сайтов, а не на реализации обходных решений, учитывающих характерные особенности браузеров. В 2013 году более половины из миллиона самых посещаемых веб-сайтов использовали jQuery. Аналогично библиотека jQuery UI, основанная на библиотеке jQuery, стала самым популярным источником виджетов пользовательского интерфейса.

С ростом популярности у разработчиков jQuery стало появляться искушение расширить библиотеку, чтобы с ее помощью можно было решать практически любые проблемы, с которыми приходится сталкиваться программистам. Однако добавление новых возможностей в ядро jQuery влечет за собой увеличение объема кода, загружаемого посетителями веб-сайтов, независимо от того, используются эти возможности на данном сайте или нет. Создание такой огромной, монолитной библиотеки неизбежно привело бы к ухудшению производительности в угоду удобству веб-разработчиков, что является далеко не самым лучшим компромиссом.

Чтобы предотвратить «разбухание» исходного кода, разработчиками jQuery была выработана философия, согласно которой в библиотеку помещается только наиболее востребованная функциональность и предоставляется возможность ее расширения. Удивительная экосистема расширений для jQuery продолжала расти все эти годы, движимая потребностью разработчиков внести свой вклад и стремлением поделиться своими наработками с обширным сообществом пользователей jQuery. Успех jQuery в значительной степени обусловлен этими благородными порывами, и команда проекта jQuery всячески способствует этому, предоставляя площадки для размещения кода, такие как сайт [plugins.jquery.com](http://plugins.jquery.com).

Кит Вуд (Keith Wood) прекрасно подходит на роль наставника по расширению библиотеки jQuery. Он является завсегдатаем форума jQuery Forum ([forum.jquery.com](http://forum.jquery.com)), где пользуется заслуженным авторитетом благодаря своим исчерпывающим ответам и решениям проблем, с которыми сталкиваются разработчики. Кроме того, его авторитет подкрепляется разработанными им популярными расшире-

ниями для jQuery. То есть Кит не только обладает обширными практическими знаниями в области расширения jQuery, но и понимает, какие особенности jQuery заслуживают подробного описания, а не мимолетного упоминания.

Эта книга детально описывает практически все аспекты расширения библиотеки jQuery, независимо от того, создаются эти расширения для личного использования или для коммерческого распространения. Наиболее простыми являются расширения методов библиотеки jQuery Core, но в книге также пристальное внимание уделяется расширениям для библиотеки виджетов jQuery UI, которые часто создаются для улучшения визуального представления информации. Детальность документации с описанием фабрики виджетов в библиотеке jQuery UI часто оказывается недостаточной, что еще больше увеличивает ценность соответствующих глав.

Мне особенно приятно отметить, что Кит уделил достаточное внимание модульным тестам. Кто-то может посчитать создание исчерпывающего комплекта модульных тестов напрасной тратой времени, но такое мнение обычно держится до момента, когда спустя несколько месяцев безобидное, казалось бы, изменение в расширении вынуждает всю команду тратить долгие часы на отладку на действующем сайте, под непрерывным потоком жалоб от пользователей. Модульные тесты не способны выявить все возможные ошибки, но они позволяют убедиться в работоспособности и предотвратить наиболее очевидные проблемы, которые легко могут пропустить нетерпеливые разработчики при тестировании вручную.

Независимо от причин, которые подтолкнули вас к изучению темы расширения возможностей jQuery, подумайте о том, чтобы поделиться своими наработками с сообществом, если вам покажется, что они смогут принести пользу кому-нибудь. Это полностью соответствует философии jQuery. Делясь своими знаниями, вы не только помогаете другим, но и зарабатываете профессиональный авторитет.

Дэйв Метвин (Dave Methvin),  
президент jQuery Foundation

# Вступление

Впервые я столкнулся с библиотекой jQuery в начале 2007 года и сразу же понял, насколько она проста и удобна в использовании. С ее помощью мне быстро удалось реализовать поиск элементов, их сокрытие и отображение. Затем я попробовал задействовать некоторые сторонние расширения и обнаружил, что они сильно отличаются своей практической ценностью и удобством использования.

Мне посчастливилось включиться в разработку расширения, ставшего впоследствии одним из самых популярных в сообществе jQuery. Я наткнулся на виджет Clean Calendar Марка Грабански (Marc Grabanski) на JavaScript, который он преобразовал в расширение для jQuery, и мне понравился его интерфейс ввода даты. Я начал экспериментировать с ним, добавлять новые особенности и попутно исследовать возможности jQuery. Результаты своих экспериментов я предложил Марку включить в его расширение. Так началась наша совместная работа над расширением, продолжавшаяся в течение следующих нескольких лет.

К тому моменту расширение Calendar было переименовано в DatePicker и выбрано командой проекта jQuery UI в качестве основы для виджета выбора даты.

Начиная с того момента я занялся разработкой других расширений, в которых испытывал потребность или к которым у меня возникал интерес. В числе моих расширений, завоевавших популярность, можно назвать: альтернативное расширение DatePicker, позволяющее выбирать диапазон дат или несколько отдельных дат; расширение Calendars с поддержкой календарей, отличных от григорианского; расширение Countdown, отображающее время, оставшееся до указанной даты и времени; и расширение SVG Integration, позволяющее взаимодействовать с SVG-элементами на веб-страницах. В процессе создания этих расширений я хорошо изучил язык JavaScript и библиотеку jQuery, а также тонкости создания расширений для последней.

Разработка расширений – идеальный способ зафиксировать функциональные возможности в форме программных компонентов многократного пользования и обеспечить простоту их внедрения в другие веб-страницы. Это подталкивает вас гораздо тщательнее тестировать свой код и гарантировать непротиворечивость его поведения в любых окружениях.

За прошедшие годы библиотека jQuery значительно выросла в размерах и в количестве функциональных возможностей, но она оста-

лась верной своей цели – максимально упростить труд разработчика. Разрастающееся сообщество разработчиков расширений – яркое свидетельство, что расширяемая архитектура для библиотеки jQuery была выбрана правильно. Я надеюсь, что взгляд изнутри на разработку расширений, представленный в этой книге, позволит вам максимально использовать возможности jQuery в своих проектах.

# Об этой книге

jQuery – самая широко используемая в Сети библиотека JavaScript, предлагающая массу возможностей и существенно облегчающая разработку веб-приложений. Но она концентрируется на предоставлении только наиболее востребованных и часто используемых функциональных возможностей и не в состоянии охватить все, что может вам потребоваться. Вы можете сами встраивать дополнительные функции в свои веб-страницы, но если вдруг обнаружится, что один и тот же код повторяется от страницы к странице, возможно, пора подумать о создании расширения для jQuery.

Расширение позволяет заключить программный код в единственный модуль, который затем можно будет применять к любому количеству веб-страниц. Преимущество единой библиотеки расширений заключается в уменьшении затрат на тестирование и сопровождение, а также в единстве внешнего вида и поведения всех страниц вашего сайта.

Библиотека jQuery изначально проектировалась с поддержкой расширений, чтобы дать им возможность стать полноправными членами окружения jQuery и использоваться наряду со встроенной функциональностью. Данная книга расскажет вам, как написать расширение, легко интегрирующееся с библиотекой, не порождающее конфликтов с другими расширениями и предоставляющее гибкое и надежное решение некоторой задачи.

## Кому адресована эта книга?

Эта книга рассказывает о расширении библиотеки jQuery и о создании модулей расширений (плагинов). Читателем может быть технический специалист, желающий узнать, что можно расширить в библиотеке jQuery, чтобы наладить создание модулей многократного пользования для своих проектов. Или им может быть веб-разработчик, стремящийся освоить тонкости разработки надежного кода для jQuery. Или сторонний разработчик расширений, желающий создать плагины для передачи сообществу jQuery.

Учитывая особенности целевой аудитории, предполагается, что вы имеете некоторое знакомство с jQuery. Вы, как ожидается, умеете с помощью jQuery выбирать элементы и выполнять операции над их свойствами, отображать или скрывать элементы и подключать к ним обработчики событий. Вы должны иметь опыт применения существ-

вующих сторонних расширений для добавления функциональности в свои страницы.

Не имеющим опыта работы с библиотекой jQuery я рекомендую предварительно прочитать книгу «jQuery in Action, Second Edition» Бера Бибо (Bear Bibeault) и Йегуды Каца (Yehuda Katz) (Manning, 2010)<sup>1</sup>.

jQuery – это библиотека на JavaScript, поэтому также предполагается, что вы знакомы с этим языком программирования. Основную часть расширений составляет код на JavaScript, содержащий вызовы методов из библиотеки jQuery или обращения к точкам интеграции. В программном коде часто используются такие конструкции, как анонимные функции, тернарные операторы и даже замыкания. Поэтому будет здорово, если эти термины знакомы вам. В противном случае, возможно, вам стоит сначала освежить свои знания языка JavaScript.

Подробное описание языка JavaScript можно найти в книге Джона Резига (John Resig) и Бера Бибо (Bear Bibeault) «Secrets of the JavaScript Ninja» (Manning 2012).

## Краткое содержание

Книга «Расширение библиотеки jQuery» делится на четыре части. Часть I (главы 1–3) охватывает простые расширения, добавляющие в библиотеку jQuery новые возможности. Часть II (главы 4–7) описывает наиболее удачные подходы к реализации расширений и функций. Часть III (главы 8–10) рассматривает вопросы расширения библиотеки jQuery UI, используемой для создания пользовательского интерфейса. Часть IV (главы 11–14) охватывает множество других немаловажных тем: анимационные эффекты, применение технологии Ajax, обработка событий и расширение Validation, которое не является частью jQuery, но играет важную роль.

- Глава 1 коротко рассказывает об истории jQuery и обсуждает возможные направления расширения библиотеки.
- Глава 2 охватывает модули, составляющие jQuery, и более подробно рассказывает о возможностях их расширения. Затем демонстрирует основные приемы разработки расширений на примере создания простого плагина.
- Глава 3 показывает, как расширять селекторы jQuery, с помощью которых выполняется поиск элементов веб-страниц.

---

<sup>1</sup> Бер Бибо, Йегуда Кац. jQuery. Подробное руководство по продвинутому JavaScript. – 2-е изд. – ISBN: 978-5-93286-201-8. – Символ-Плюс, 2011.

- Глава 4 отступает на шаг назад и рассказывает о передовых приемах разработки, помогающих создавать надежные и полезные расширения.
- Глава 5 демонстрирует создание расширения коллекций на основе принципов, описанных в предыдущей главе. Расширения коллекций воздействуют сразу на множество элементов страницы.
- Глава 6 рассматривает функциональные расширения, реализующие дополнительные возможности, не связанные с какими-то конкретными элементами, например локализация приложения или обработка блоков `cookie`.
- Глава 7 обсуждает приемы тестирования и упаковки, гарантирующие корректную работу расширений и упрощающие их получение и использование. В ней также описывается, как документировать и демонстрировать расширение, чтобы пользователи могли применять все его возможности.
- Глава 8 показывает, как на основе библиотеки `jQuery UI` создавать расширения коллекций, позволяющие объединять компоненты `jQuery UI` в единый виджет.
- Глава 9 объясняет, как с помощью модуля `jQuery UI Mouse` реализовать рисование с помощью мыши внутри собственного расширения.
- Глава 10 завершает часть книги, посвященную библиотеке `jQuery UI`, знакомством с возможностями создания визуальных эффектов и корректировки частоты изменения анимируемых свойств.
- Глава 11 показывает, как передавать значения для анимируемых свойств, не являющиеся простыми числовыми значениями, на примере CSS-свойства `background-position`.
- Глава 12 рассказывает о возможностях поддержки технологии `Ajax` в библиотеке `jQuery` и показывает, как расширить их с помощью своих фильтров, транспортов и преобразователей.
- Глава 13 обсуждает особенности фреймворка обработки событий в `jQuery` и как его использовать для создания новых и расширения существующих событий `jQuery`.
- Глава 14 показывает, как расширить плагин `Validation` и добавить в него дополнительные правила проверки, которые могли бы применяться к отдельным элементам наряду со встроенными.



## Соглашения об оформлении программного кода и комплект загружаемых примеров

В этой книге приводится множество листингов программного кода на JavaScript, а также разметки HTML и правил CSS. Исходный код в листингах и в тексте будет оформляться моноширинным шрифтом, чтобы его можно было отличить от обычного текста. Имена переменных и функций в тексте также будут оформляться подобным образом. **Жирным моноширинным шрифтом** будут выделяться ключевые фрагменты кода, обычно имена функций или переменных. Некоторые листинги подверглись реформатированию с целью уместить их по ширине книжной страницы. Большинство листингов включают дополнительные аннотации для выделения наиболее важных частей. Во многих случаях нумерованным указателям в коде соответствуют дополнительные тексты в следующем ниже тексте.

jQuery и jQuery UI – это открытые библиотеки, распространяемые на условиях лицензии MIT<sup>1</sup> и могут быть загружены на веб-сайтах <http://jquery.com/> и <http://jqueryui.com/> соответственно. Исходный код примеров для этой книги доступен для загрузки на веб-сайте издательства Manning: <http://www.manning.com/ExtendingjQuery>.

### Автор в Сети

Одновременно с покупкой книги «Расширение библиотеки jQuery» вы получаете бесплатный доступ к частному веб-форуму, организованному издательством Manning Publications, где можно оставлять комментарии о книге, задавать технические вопросы, а также получать помощь от автора и других пользователей. Чтобы получить доступ к форуму и зарегистрироваться на нем, откройте в веб-браузере страницу <http://www.manning.com/ExtendingjQuery>. Здесь описывается, как попасть на форум после регистрации, какие виды помощи доступны и правила поведения на форуме.

Издательство Manning обязуется предоставить своим читателям место встречи, где может состояться содержательный диалог между отдельными читателями и между читателями и автором. Но со стороны автора отсутствуют какие-либо обязательства уделять форуму какое-то определенное внимание – его присутствие на форуме оста-

---

<sup>1</sup> Текст лицензии можно найти по адресу: <https://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt>. (См. также [http://ru.wikipedia.org/wiki/Лицензия\\_MIT](http://ru.wikipedia.org/wiki/Лицензия_MIT). – *Прим. перев.*)

ется добровольным (и неоплачиваемым). Мы предлагаем задавать автору стимулирующие вопросы, чтобы его интерес не угасал!

## **Об авторе**

Кит Вуд (Keith Wood) – разработчик с 30-летним стажем, использующий библиотеку jQuery с 2007 года. Им было написано более 20 расширений – включая оригинальный плагин Daterangepicker, World Calendar and Daterangepicker, Countdown и SVG, – которые он передал сообществу пользователей jQuery. Он часто отвечает на вопросы, касающиеся библиотеки jQuery, на форумах и в 2012 году вошел в пятерку самых активных участников.

На своей основной работе он занимается созданием веб-приложений с использованием Java/J2EE на стороне сервера и jQuery на стороне клиента. Проживает в Австралии, в городе Сидней, со своей супругой Тресили (Tresialee). В свободное от работы время увлекается танцами.

## Простые расширения

Будучи наиболее широко используемой библиотекой JavaScript, jQuery предлагает массу функций, упрощающих разработку пользовательских интерфейсов. Но вы можете сделать библиотеку jQuery еще лучше, расширяя ее дополнительными возможностями многократного пользования.

Глава 1 коротко рассказывает об истории развития jQuery, а затем показывает, какие ее элементы можно расширить. Она завершается несколькими примерами имеющихся расширений, демонстрирующими широту возможностей.

В главе 2 вы найдете описание архитектуры jQuery и доступных точек интеграции, с подробным описанием каждой из них. Затем мы познакомимся с процессом разработки простого расширения, которое можно будет немедленно включить в работу.

Простейшими расширениями, которые вам может понадобиться создавать, являются селекторы – строительные блоки, на которых основан механизм поиска элементов для выполнения последующих операций над ними. О селекторах рассказывается в главе 3, где также приводятся многочисленные примеры создания собственных селекторов.

# Глава 1

## Расширения для jQuery

Эта глава охватывает следующие темы:

- происхождение и назначение jQuery;
- что в библиотеке jQuery доступно для расширения;
- примеры существующих расширений.

В настоящее время jQuery является самой распространенной библиотекой JavaScript в Сети. Она предлагает множество функциональных возможностей, упрощающих разработку пользовательских интерфейсов, таких как обход дерева объектной модели документа HTML (Document Object Model, DOM) в поисках элементов для последующей обработки и применения анимационных эффектов к ним. Кроме того, разработчики jQuery понимают, что невозможно (да и не нужно) предусмотреть все необходимое, и предусмотрели точки интеграции, посредством которых можно добавлять в библиотеку новые функциональные возможности. Такая дальновидность способствовала росту популярности jQuery.

В этой книге я расскажу, как расширять различные аспекты jQuery, чтобы упростить возможность повторного использования расширений и их сопровождение. Помимо стандартных расширений, воздействующих на коллекции элементов веб-страниц, есть возможность создавать собственные селекторы, вспомогательные функции, анимационные эффекты, дополнительные механизмы обработки для использования в технологии Ajax, собственные события и правила проверки данных. Я покажу, как выполнять тестирование, упаковку и документирование кода, чтобы другие разработчики смогли максимально использовать его возможности.

### 1.1. История развития jQuery

На веб-сайте проекта библиотека jQuery характеризуется как «быстрая, маленькая и богатая возможностями библиотека JavaScript. Она позволяет выполнять обход элементов документа HTML, производя

операции над ними, обрабатывать события, воспроизводить анимационные эффекты и существенно упрощает использование технологии Ajax, предоставляя простой в использовании API, одинаковый для самых разных браузеров» (<http://jquery.com>).

Это библиотека функций JavaScript, обеспечивающих простой доступ к HTML DOM, его исследование и изменение, дающих возможность делать веб-страницы более динамичными и выразительными в соответствии с парадигмой Web 2.0. Главными особенностями библиотеки являются:

- выбор элементов с использованием CSS-подобного синтаксиса;
- возможность обхода элементов;
- поддержка операций с элементами, включая удаление и изменение содержимого и атрибутов;
- обработка событий, в том числе нестандартных;
- визуальные и анимационные эффекты;
- поддержка Ajax;
- возможность расширения функциональности (что является темой этой книги);
- различные вспомогательные функции;
- поддержка браузеров разных типов и сокрытие несовместимостей между ними.

jQuery является открытой и свободно распространяемой библиотекой. В настоящее время она распространяется на условиях лицензии MIT (<http://jquery.org/license/><sup>1</sup>). Прежние ее версии распространялись также на условиях лицензии GNU General Public License, Version 2<sup>2</sup>.

### 1.1.1. Происхождение

Первая версия jQuery была создана Джоном Резигом (John Resig) и анонсирована в январе 2006 года на конференции BarCamp NYC<sup>3</sup>. Он случайно наткнулся на библиотеку Behaviour, написанную Беном Ноланом (Ben Nolan), и оценил потенциал заложенных в нее идей – использование CSS-подобных селекторов для привязки различных

<sup>1</sup> [http://ru.wikipedia.org/wiki/Лицензия\\_MIT](http://ru.wikipedia.org/wiki/Лицензия_MIT). – Прим. перев.

<sup>2</sup> Перевод текста лицензии GNU GPL v2 можно найти по адресу: <http://jxself.org/translations/gpl-2.ru.shtml>. – Прим. перев.

<sup>3</sup> Джон Резиг (John Resig), «BarCampNYC Wrap-up», <http://ejohn.org/blog/barcampnyc-wrap-up/>.

функций JavaScript к элементам DOM. Но Джону не понравилась многословность синтаксиса и отсутствие иерархических селекторов<sup>1</sup>. Предложенный им синтаксис и последующая реализация легли в основу библиотеки jQuery.

В листинге 1.1 демонстрируется код, использующий библиотеку Behaviour для подключения обработчика события щелчка мышью ко всем элементам li, находящимся в элементе с атрибутом id="example"; обработчик события удаляет элемент, на котором выполнен щелчок. В листинге 1.2 демонстрируется аналогичный код, использующий библиотеку jQuery.

### Листинг 1.1. Пример использования библиотеки Behaviour

```
Behaviour.register({
  '#example li': function(e){
    e.onclick = function(){
      this.parentNode.removeChild(this);
    }
  }
});
```

### Листинг 1.2. Эквивалентный код, использующий библиотеку jQuery

```
$('#example li').bind('click', function(){
  $(this).remove();
});
```

Как было выбрано имя jQuery? Первоначально библиотеке было дано имя jSelect, отражающее возможность выбирать элементы веб-страницы. Но когда Джон выполнил поиск по этому имени, он обнаружил, что оно уже занято, и присвоил библиотеке имя jQuery<sup>2</sup>.

## 1.1.2. Развитие

С момента первоначального анонса было выпущено множество версий библиотеки jQuery, которые перечислены в табл. 1.1 (здесь представлены не все версии). С годами она выросла весьма существенно как с точки зрения функциональности, так и с точки зрения размеров.

Несмотря на существенный рост размеров библиотеки jQuery, при минификации кода (удалении ненужных комментариев и пробельных символов) ее размер уменьшается в три раза (после минифика-

---

<sup>1</sup> Джон Резиг (John Resig), «Selectors in Javascript», <http://ejohn.org/blog/selectors-in-javascript/>.

<sup>2</sup> Комментарии Джона Резига, «BarCampNYC Wrap-up», <http://ejohn.org/blog/barcampnyc-wrap-up/>.

**Таблица 1.1. Версии библиотеки jQuery (не все)**

Версия	Дата выпуска	Размер, Кбайт	Примечания
1.0	26.08.2006	44,3	Первая стабильная версия
1.0.4	12.12.2006	52,2	Последняя исправленная версия 1.0
1.1	14.01.2007	55,6	Улучшение производительности селекторов
1.1.4	23.08.2007	65,6	jQuery могла быть переименована
1.2	10.09.2007	77,4	
1.2.6	26.05.2008	97,8	
1.3	13.01.2009	114	Встроен новый механизм селекторов Sizzle, добавлено динамическое управление событиями, перестроен механизм управления событиями
1.3.2	19.02.2009	117	
1.4	13.01.2010	154	Улучшение производительности, расширение поддержки технологии Ajax
1.4.1	25.01.2010	156	Добавлены методы <code>height()</code> , <code>width()</code> и <code>parseJSON()</code>
1.4.2	13.02.2010	160	Улучшение производительности, добавлен метод <code>delegate()</code>
1.4.3	14.10.2010	176	Переписан модуль CSS, добавлена обработка метаданных
1.4.4	11.11.2010	178	
1.5	31.01.2011	207	Управление отложенными обратными вызовами, переписан модуль Ajax, улучшена производительность механизма обхода дерева DOM
1.5.2	31.03.2011	214	
1.6	02.05.2011	227	Существенно увеличена производительность функций <code>attr()</code> и <code>val()</code> , добавлена функция <code>prop()</code>
1.6.4	12.09.2011	232	
1.7	03.11.2011	243	Новые функции в Event API: <code>on()</code> и <code>off()</code> , улучшена производительность механизма делегирования событий
1.7.2	21.03.2012	246	
1.8.0	09.08.2012	253	Переписан механизм селекторов Sizzle, переработан механизм поддержки анимации, улучшена модульность библиотеки
1.8.3	13.11.2012	261	
1.9.0	14.01.2013	261	Чистка кода, подготовка к выпуску версии jQuery 2.0
1.9.1	04.02.2013	262	Исправление ошибок и регрессий
2.0.0	18.04.2013	234	Исключена поддержка IE 6–8
1.10.0	24.05.2013	267	Синхронизация возможностей с веткой 2.x
1.10.2	03.07.2013	266	
2.0.3	03.07.2013	236	

ции размер последней версии уменьшается до 91 Кбайта). Если минифицированная версия возвращается веб-сервером в формате gzip, ее размер уменьшается еще примерно в три раза – ориентировочно до 32 Кбайт в последней версии. При использовании сети доставки содержимого (Content Delivery Network, CDN) этот файл может сохраняться в кэше клиента, избавляя от необходимости повторно загружать его.

### Использование CDN

Чтобы загрузить библиотеку jQuery с одного из серверов CDN, включите один из тегов `script`, представленных в этой врезке. Вам может потребоваться изменить номер загружаемой версии, выбрав наиболее подходящую для ваших требований.

#### Использование сети CDN, поддерживаемой компанией MediaTemple

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js">
</script>
```

Вы можете также включить загрузку расширения jQuery Migration с этого же сайта, который поможет выполнить переход с более старых версий jQuery на версию jQuery 1.9 или выше.

```
<script
  src="http://code.jquery.com/jquery-migrate-1.1.1.min.js">
</script>
```

#### Использование сети CDN, поддерживаемой компанией Google<sup>1</sup>

```
<script
  src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.
  min.js">
</script>
```

В сети CDN компании Google доступны все версии jQuery, но эта сеть неподконтрольна разработчикам jQuery, и более свежие версии могут появляться в ней с некоторым опозданием.

#### Использование сети CDN, поддерживаемой компанией Microsoft<sup>2</sup>

```
<script
  src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.9.1.min.js">
</script>
```

В сети CDN компании Microsoft CDN доступны все версии jQuery, но эта сеть неподконтрольна разработчикам jQuery, и более свежие версии могут появляться в ней с некоторым опозданием.

---

<sup>1</sup> Список библиотек в сети CDN компании – руководство разработчика» <https://developers.google.com/speed/libraries/devguide#jquery>.

<sup>2</sup> ASP.NET, «Сеть доставки содержимого компании Microsoft» <http://www.asp.net/ajaxlibrary/cdn.ashx>.



В настоящее время jQuery включает механизм селекторов *Sizzle*, реализующий фундаментальную возможность поиска элементов DOM. Всякий раз, когда это возможно, механизм *Sizzle* делегирует обработку селекторов браузеру, но при необходимости возвращается к собственной реализации на JavaScript, чтобы обеспечить равноценную поддержку во всех основных браузерах.

### 1.1.3. Современное состояние дел

jQuery стала самой популярной библиотекой JavaScript в Интернете и используется на значительном количестве веб-сайтов. Она официально поддерживается компанией Microsoft и распространяется в составе комплекта продуктов Visual Studio. Компания BuiltWith сообщает, что более 60% из 10 000 самых посещаемых веб-сайтов используют jQuery, и более 50% – из миллиона<sup>1</sup>. Компания W3Techs сообщает, что jQuery используется на 55% всех веб-сайтов и на 90% сайтов, где вообще используется хоть какая-нибудь библиотека JavaScript<sup>2</sup>.

Сообщество разработчиков расширений постоянно ширится, и многие из них открывают свободный доступ к своему коду в духе самой библиотеки jQuery. Вы можете самостоятельно поискать необходимые вам модули или воспользоваться постоянно обновляемым «официальным» репозиторием расширений для jQuery (<http://plugins.jquery.com>). Некоторые расширения просто великолепны – содержат надежный код, сопровождаются исчерпывающей документацией и примерами. Другие не так хороши, сложны в использовании, содержат ошибки и/или плохо документированы. Когда вы прочтаете эту книгу и будете следовать описываемым в ней принципам, ваши расширения, безусловно, попадут в первую категорию.

Активная жизнь также кипит на форумах, посвященных jQuery (<https://forum.jquery.com>), где можно найти более 250 000 ответов на более чем 110 000 вопросов. На форумах вы найдете специальные разделы, где обсуждаются вопросы создания и использования расширений.

Разработка библиотеки jQuery в настоящее время продолжается под эгидой организации jQuery Foundation (<http://jquery.org>). Она

---

<sup>1</sup> Статистика использования jQuery от компании BuiltWith: <http://trends.builtwith.com/javascript/jquery>.

<sup>2</sup> Статистика использования библиотек JavaScript на сайте компании W3Techs: [http://w3techs.com/technologies/overview/javascript\\_library/all](http://w3techs.com/technologies/overview/javascript_library/all).

была сформирована в сентябре 2009 года с целью поддержки всех проектов линейки jQuery, включая jQuery Core, jQuery UI, jQuery Mobile, Sizzle и QUnit. Финансовую основу этой поддержки составляют взносы и пожертвования сообщества jQuery.

## 1.2. Расширение jQuery

Если библиотека jQuery обладает столь широкими функциональными возможностями, зачем еще больше расширять их? Чтобы удержать объем кода jQuery в разумных пределах, в ядро библиотеки включены только наиболее востребованные и универсальные функции (хотя до сих пор не утихают дебаты о том, что считать наиболее востребованным). Базовые функции доступа к элементам, обработка событий, анимация и поддержка Ajax – вот что требуется большинству пользователей, при этом сохраняется возможность добавлять более специализированные функции.

К счастью, команда проекта jQuery прекрасно понимает, что ядро jQuery не может вместить все, что только может потребоваться, поэтому было создано множество точек интеграции для расширения функциональности jQuery и использования существующей инфраструктуры и возможностей.

Упаковка реализованного расширения для jQuery в виде плагина позволит вам повторно использовать новые функциональные возможности на других веб-страницах. В результате вам придется поддерживать единственную копию расширения, и любые последующие улучшения в нем будут немедленно вступать в силу везде, где оно используется. Вы можете организовать тестирование своего плагина в изолированном окружении и при определенных условиях, гарантирующих нормальную работу расширения.

### 1.2.1. Что доступно для расширения?

Несмотря на значительное разнообразие функциональных возможностей, предоставляемых ядром библиотеки, всегда можно найти массу аспектов, функциональность которых можно было бы расширить. Те аспекты, которые охватываются в этой книге, перечислены в следующих разделах.

#### *Селекторы и фильтры*

Селекторы и фильтры jQuery позволяют идентифицировать и отбирать элементы веб-страницы для выполнения последующих опера-

ций с ними. Несмотря на наличие стандартных селекторов, отбирающих узлы по именам, атрибутам `id` и `class`, иногда бывает желательно иметь дополнительные селекторы псевдоклассов (расширяющих набор псевдоклассов CSS), чтобы можно было фильтровать прежде выбранное множество элементов кратким и непротиворечивым способом. Можно также добавлять фильтры, которые принимают коллекцию предварительно отобранных элементов и фильтруют ее, опираясь на позиции отдельных элементов в этой коллекции. Создание селекторов данных типов описывается в главе 3.

Создавая собственные селекторы, можно реализовать процедуру выбора в одном месте и с легкостью использовать везде, где угодно, гарантируя единообразие реализации во всех проектах. Такой подход также упрощает сопровождение селектора и немедленное применение любых исправлений или расширений ко всем веб-страницам, где он используется.

### ***Расширения коллекций***

*Расширения коллекций* – это функции, которые могут применяться к коллекциям элементов, полученным с помощью селекторов. Именно эти функции подразумеваются многими под термином *расширение для jQuery*, и именно они образуют самую многочисленную группу расширений. Возможности расширений коллекций ограничиваются только вашей фантазией. Им подвластно все – от простого изменения атрибутов элементов до изменения поведенческих характеристик за счет обработки событий в этих элементах и полной замены оригинальных компонентов альтернативной реализацией.

В главе 4 представлено несколько правил, которыми желательно руководствоваться при создании расширений, а в главе 5 описывается инфраструктура поддержки расширений, которую я использую для создания собственных расширений, и демонстрируется ее соответствие упомянутым выше правилам. Правила аккумулируют в себе опыт, накопленный при создании расширений, они помогают легко интегрировать собственные расширения с библиотекой jQuery и уменьшают вероятность конфликтов с внешним кодом.

Ключевым пунктом в процессе разработки собственных расширений является их тестирование, и применение инструментов модульного тестирования позволит вам без особых усилий тестировать свой код и убеждаться в его работоспособности. Как только код будет готов к выпуску, его необходимо упаковать в дистрибутив, чтобы другие разработчики могли легко получить его и интегрировать в свои

проекты. Вы должны также создать веб-страницу, демонстрирующую возможности расширения, чтобы позволить предполагаемым пользователям познакомиться с его возможностями и особенностями. И не забудьте про документацию, описывающую все аспекты применения вашего расширения, которая даст возможность другим максимально использовать его потенциал. Все эти стороны разработки расширений освещаются в главе 7.

### ***Расширения-функции***

*Расширения-функции* – это вспомогательные функции, которые не выполняют операции непосредственно с коллекциями элементов. Они реализуют дополнительные возможности и обычно используют механизмы библиотеки jQuery для выполнения своей работы. В главе 6 подробно рассказывается, как создавать такие функции.

В качестве примеров подобных расширений можно привести вывод отладочных сообщений в консоль или извлечение и изменение значений cookie для веб-страницы. Реализуя такие возможности в виде расширений jQuery, вы предоставляете пользователям знакомый способ вызова кода и уменьшаете вероятность конфликтов с внешним кодом. Некоторые правила, упоминавшиеся выше, применимы и к расширениям этого вида, как и рекомендации, касающиеся тестирования, упаковки, демонстрации и документирования.

### ***Виджеты jQuery UI***

*jQuery UI* – «это организованный набор инструментов взаимодействия с пользовательским интерфейсом, эффектов, виджетов и тем оформления, опирающихся на основную библиотеку jQuery» (<http://jqueryui.com/>). Библиотека jQuery UI определяет инфраструктуру, позволяющую создавать виджеты, действующие непротиворечивым образом и способные использовать огромное количество тем оформления пользовательского интерфейса. В главе 8 рассказывается об инфраструктуре поддержки виджетов и особенностях ее использования для создания собственных визуальных компонентов.

Инфраструктура виджетов jQuery UI также следует правилам, описываемым в главе 4, и предоставляет функциональность, общую для всех виджетов jQuery UI. Принимая эту инфраструктуру в качестве основы для своих расширений, вы автоматически получаете весь ее потенциал и возможность сконцентрироваться на создании функциональности, уникальной для вашего виджета. Если вы задействуете в своем виджете классы, определенные в ThemeRoller, он ав-

томатически будет следовать общему стилю оформления с другими компонентами jQuery UI и изменяться при применении новой темы.

Некоторые виджеты jQuery UI предусматривают возможность взаимодействий с указателем мыши, и команда проекта jQuery UI понимает всю важность таких взаимодействий. Наследуя в своем виджете модуль jQuery UI Mouse вместо базового модуля Widget, вы автоматически получаете поддержку операций с мышью, позволяющую настраивать все аспекты взаимодействий, и вы снова можете сконцентрироваться на реализации особенностей, уникальных для вашего виджета. Глава 9 описывает, как задействовать модуль Mouse при создании виджета, функционирование которого основано на использовании мыши.

### **Эффекты jQuery UI**

В состав jQuery UI входит также множество эффектов, которые могут применяться к элементам страницы. Эффекты, такие как `blind`, `clip`, `fold` и `slide`, можно использовать для отображения или сокрытия элементов. Некоторые эффекты, такие как `highlight` и `pulsate`, предназначены для привлечения внимания к элементу. Вы можете определить собственные эффекты и применять их к элементам наряду со стандартными. Глава 10 демонстрирует порядок создания новых эффектов.

### **Анимация свойств**

В jQuery имеется механизм *анимации*, который можно задействовать для изменения любых атрибутов элементов, принимающих обычные числовые значения. Он позволяет организовать поэтапное изменение атрибута от одного значения до другого, управлять продолжительностью изменения и величиной одного шага. Но если значение, которое требуется изменять, не является простым числовым значением, вам придется реализовать необходимую функциональность самостоятельно. Например, в составе jQuery UI имеется модуль, дающий возможность изменять цвет от одного значения до другого. В главе 11 мы реализуем анимацию для атрибута, имеющего составное значение.

### **Поддержка AJAX**

Поддержка технологии *Ajax* в библиотеке jQuery является одним из очевидных ее преимуществ. Она позволяет легко загружать удаленные данные и обрабатывать их. Обращаясь к механизму Ajax,

вы можете указать тип данных, которые готов принять ваш обработчик успешного завершения вызова: простой текст, HTML, XML, JSON. Преобразование потока байтов, поступающих от удаленной стороны, будет выполнено библиотекой автоматически. Вы можете добавлять собственные преобразования для поддержки специализированных форматов простым указанием типа данных, который должен быть возвращен. Глава 12 обсуждает порядок расширения механизма поддержки Ajax для обработки распространенных форматов файлов.

### ***Обработка событий***

Средства обработки событий в библиотеке jQuery дают возможность подключать к элементам множество обработчиков для обработки действий пользователя, системных и собственных событий. В jQuery имеется несколько точек входа для определения нестандартных событий и их возбуждения, благодаря чему обеспечивается тесная интеграция с существующей функциональностью. Глава 13 описывает реализацию нового события с целью упрощения взаимодействий с мышью.

### ***Правила проверки***

Расширение Validation, написанное Йерном Зафферером (Jörn Zaefferer), широко используется для проверки ввода пользователя на стороне клиента перед отправкой значений на сервер. Несмотря на то что это расширение не входит в ядро библиотеки jQuery, оно также предоставляет точки для подключения дополнительных расширений, благодаря которым можно определять собственные правила проверки и включать их в существующую процедуру обработки. Глава 14 иллюстрирует, как создавать собственные правила проверки и передавать их встроенному механизму для выполнения.

## **1.3. Примеры расширений**

В Интернете можно найти сотни расширений для библиотеки jQuery, добавляющих новые функциональные возможности в веб-страницы. Их количество – самое веское свидетельство широты возможностей jQuery и простоты их использования, а также дальновидности разработчиков, предоставивших точки подключения расширений к библиотеке. Я не смогу охватить все расширения в этой книге, но в следующих разделах покажу уровень имеющихся возможностей.