

# PHP и jQuery

ДЛЯ ПРОФЕССИОНАЛОВ

# Pro PHP and jQuery

Jason Lengstorf

Apress®

# PHP и jQuery

## ДЛЯ ПРОФЕССИОНАЛОВ

Джейсон Ленгсторф



Москва • Санкт-Петербург • Киев  
2011

ББК 32.973.26-018.2.75

Л44

УДК 681.3.07

Издательский дом “Вильямс”

Главный редактор *С.Н. Тригуб*

Зав. редакцией *В.Р. Гинзбург*

Перевод с английского и редакция канд. хим. наук А.Г. Гузикевича

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:  
info@williamspublishing.com, <http://www.williamspublishing.com>

**Ленгсторф, Джейсон.**

Л44 РНР и jQuery для профессионалов. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2011. — 352 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1693-8 (рус.)

**ББК 32.973.26-018.2.75**

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства APress, Berkeley, CA.

Authorized translation from the English language edition published by APress, Copyright © 2010 by Jason Lengstorf.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the publisher.

Russian language edition is published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2011.

Научно-популярное издание

**Джейсон Ленгсторф**

## **РНР и jQuery для профессионалов**

Литературный редактор *Е.Д. Давидян*

Верстка *М.А. Удалов*

Художественный редактор *В.Г. Павлютин*

Корректор *Л.А. Гордиенко*

Подписано в печать 25.10.2010. Формат 70x100/16. Гарнитура Times. Печать офсетная  
Усл. печ. л. 28,38. Уч.-изд. л. 15,1 Тираж 1500 экз. Заказ № 0000

Отпечатано по технологии CtP

в ОАО “Печатный двор” им. А. М. Горького. 197110, Санкт-Петербург, Чкаловский пр., 15

ООО “И. Д. Вильямс”, 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1693-8 (рус.)

ISBN 978-1-4302-2847-9 (англ.)

© Издательский дом “Вильямс”, 2011

© Jason Lengstorf, 2010

# Оглавление

<b>Часть I. Основные сведения о jQuery</b>	<b>13</b>
Глава 1. Введение в технологию jQuery	15
Глава 2. Распространенные операции и методы jQuery	33
<b>Часть II. Профессиональные аспекты программирования на PHP</b>	<b>83</b>
Глава 3. Объектно-ориентированное программирование	85
Глава 4. Создание календаря событий	113
Глава 5. Добавление элементов управления для создания, редактирования и удаления событий	157
Глава 6. Парольная защита критических данных и операций над ними	187
<b>Часть III. Добавление сценариев jQuery в PHP-приложения</b>	<b>219</b>
Глава 7. Улучшение пользовательского интерфейса средствами jQuery	221
Глава 8. Редактирование данных календаря средствами AJAX и jQuery	247
<b>Часть IV. Дополнительные возможности jQuery и PHP</b>	<b>289</b>
Глава 9. Проверка форм с помощью регулярных выражений	291
Глава 10. Расширение jQuery	321
<b>Предметный указатель</b>	<b>337</b>

# Содержание

Об авторе	11
О рецензенте	12
<b>Часть I. Основные сведения о jQuery</b>	<b>13</b>
<b>Глава 1. Введение в технологию jQuery</b>	<b>15</b>
К вопросу о выборе jQuery	15
Библиотеки JavaScript	15
Преимущества, обеспечиваемые jQuery	16
История возникновения jQuery	16
Настройка тестовой среды	16
Установка Firefox	17
Установка Firebug	17
Подключение jQuery к веб-страницам	19
Подключение загруженной копии библиотеки jQuery	19
Подключение копии библиотеки jQuery, хранящейся на удаленном хосте	19
Использование Google Libraries API	19
Создание тестового файла	20
Функция jQuery ()	21
Выбор DOM-элементов с использованием синтаксиса CSS	21
Резюме	32
<b>Глава 2. Распространенные операции и методы jQuery</b>	<b>33</b>
Особенности поведения сценариев jQuery	33
Методы jQuery	33
Обход DOM-элементов	34
Создание и вставка DOM-элементов	42
Доступ к CSS и атрибутам и их изменение	54
Воздействие на результирующие наборы	62
Использование анимации и других эффектов	63
Обработка событий	70
Использование элементов управления AJAX	76
Резюме	81
<b>Часть II. Профессиональные аспекты программирования на PHP</b>	<b>83</b>
<b>Глава 3. Объектно-ориентированное программирование</b>	<b>85</b>
Принципы ООП	85
Объекты и классы	86
Различия между объектами и классами	86
Структура классов	86
Определение свойств класса	87
Определение методов класса	88

Наследование классов	95
Назначение области видимости свойствам и методам	99
Создание комментариев в стиле Дос-блоков	105
Преимущества ООП в сравнении с процедурным подходом	107
Простота реализации	107
Улучшение структуры приложения	111
Легкость сопровождения	111
Резюме	112
<b>Глава 4. Создание календаря событий</b>	<b>113</b>
Планирование приложения	113
Определение структуры базы данных	113
Создание схемы класса	114
Планирование структуры папок приложения	114
Настройка среды разработки	116
Создание календаря	117
Создание базы данных	118
Класс для подключения к базе данных	118
Создание класса-оболочки для приложения	120
Добавление свойств класса	121
Создание конструктора	122
Загрузка информации о событиях	128
Вывод HTML-кода для отображения календаря и событий	135
Вывод HTML-кода для отображения подробного описания события	150
Резюме	156
<b>Глава 5. Добавление элементов управления для создания, редактирования и удаления событий</b>	<b>157</b>
Генерация формы для создания и редактирования событий	157
Добавление маркера в форму	159
Создание файла для отображения формы	161
Добавление новой таблицы стилей для средств администрирования	162
Сохранение новых событий в базе данных	165
Добавление файла, осуществляющего вызов запрошенного метода обработки формы	168
Добавление кнопки создания новых событий в основное представление	171
Добавление элементов редактирования в подробное представление событий	174
Модификация метода <code>displayEvent</code> для отображения элементов административного управления	175
Добавление административной таблицы стилей в подробное представление событий	176
Удаление событий	179
Генерация кнопки удаления события	179

## 8 Содержание

Создание метода, запрашивающего подтверждение удаления события	180
Создание файла для отображения подтверждающей формы	183
Резюме	186
<b>Глава 6. Парольная защита критических данных и операций над ними</b>	187
Создание административной таблицы в базе данных	187
Создание файла для отображения регистрационной формы	188
Создание класса <code>Admin</code>	190
Определение класса	190
Создание метода для проверки учетных данных пользователя	191
Модификация приложения для обработки отправки регистрационной формы	200
Предоставление зарегистрированному пользователю возможности завершения сеанса	203
Добавление кнопки выхода	203
Создание метода для процедуры выхода	206
Модификация приложения для обработки завершения сеанса	207
Отображение элементов административного управления	210
Отображение административных опций лишь для администраторов	210
Ограничение доступа к административным страницам	214
Резюме	217
<b>Часть III. Добавление сценариев jQuery в PHP-приложения</b>	219
<b>Глава 7. Улучшение пользовательского интерфейса средствами jQuery</b>	221
Прогрессивное улучшение приложения с помощью jQuery	221
Постановка задачи	222
Подключение jQuery к приложению	222
Создание файла инициализации JavaScript	223
Создание новой таблицы стилей для элементов, созданных jQuery	224
Создание модального окна для отображения информации о событии	226
Связывание функции с событием щелчка на ссылке названия	226
Предотвращение выполнения действия по умолчанию и добавление класса <code>active</code>	226
Извлечение строки запроса с помощью регулярных выражений	227
Создание модального окна	229
Извлечение и отображение информации о событиях с помощью AJAX	233
Добавление кнопки закрытия окна	238
Добавление эффектов в процессы создания и уничтожения модального окна	239
Резюме	246



<b>Глава 8. Редактирование данных календаря средствами AJAX и jQuery</b>	247
Открытие формы для создания событий	247
Добавление вызова AJAX для загрузки формы	248
Модификация обрабатывающего файла AJAX для загрузки формы	250
Закрытие модального окна при щелчке на кнопке Отменить	252
Сохранение новых событий в базе данных	252
Сериализация данных формы	253
Отправка сериализованных данных формы обрабатывающему файлу	254
Модификация обрабатывающего файла AJAX для обработки новых отправок	255
Добавление событий без обновления страницы	257
Десериализация данных формы	257
Создание объектов <b>Date</b>	262
Присоединение событий к календарю	266
Получение идентификатора нового события	269
Редактирование событий в модальном окне	271
Определение атрибута <b>action</b> для формы	273
Сохранение идентификатора события, если таковой существует	274
Удаление информации о событии из модального окна	276
Добавление только новых событий в календарь	277
Подтверждение удаления событий в модальном окне	280
Отображение окна подтверждения	280
Настройка обработчика события отправки формы, предназначенной для удаления события	281
Исключение события из календаря после его удаления	285
Резюме	288
<b>Часть IV. Дополнительные возможности jQuery и PHP</b>	289
<b>Глава 9. Проверка форм с помощью регулярных выражений</b>	291
Введение в регулярные выражения	291
Базовый синтаксис регулярных выражений	291
Детализация информации на основе модификаторов шаблонов	295
Использование обратных ссылок в регулярных выражениях	297
Поиск соответствий шаблону с помощью символьных классов	300
Нахождение границ слов	302
Операторы повторения	303
Обнаружение начала и конца строки	303
Использование альтернативных шаблонов	304
Использование необязательных элементов	304
Сводим все вместе	304
Проверка допустимости введенных значений даты и времени на стороне сервера	307

## 10 Содержание

Определение шаблона регулярного выражения для проверки формата даты и времени	307
Добавление метода проверки в класс <code>Calendar</code>	311
Возврат ошибки в случае недопустимости даты и времени	312
Проверка допустимости задания даты и времени на стороне клиента	316
Создание нового файла сценария JavaScript для проверки допустимости значений даты и времени	316
Включение нового файла в завершающую часть страницы	316
Предотвращение отправки формы в случае отрицательного результата проверки	317
Резюме	320
<b>Глава 10. Расширение jQuery</b>	<b>321</b>
Добавление функций в jQuery	321
Добавление функции проверки даты и времени в jQuery	321
Модификация сценария, выполняющего включение файлов	324
Модификация сценария инициализации	325
Добавление методов в jQuery	327
Создание собственного подключаемого модуля	327
Внедрение дополнения	333
Резюме	336
<b>Предметный указатель</b>	<b>337</b>

## Об авторе



**Джейсон Ленгсторф** — веб-дизайнер и разработчик, проживающий в штате Монтана. Его специализация — создание программного обеспечения для управления веб-контентом на основе PHP, MySQL, AJAX и веб-стандартов. Большую часть времени отдает своей компании Eppoi Design, интернациональный коллектив которой разрабатывает первоклассные веб-сайты. Он также управляет собственной фирмой по пошиву одежды Humblecock, пытаясь при этом выкроить время для своих хобби, таких как гольф и путешествия.

## О рецензенте



**Роберт Бан** (Robert Banh) — опытный разработчик, занимающийся программированием чуть ли не с самого детства. Специализируется на разработке веб-приложений с использованием PHP/MySQL, Zend и CodeIgniter для крупных заказчиков, таких как IBM, HP, Unisys и KLRU, а также для общественных организаций. В настоящее время работает в Техасском университете в Остине, где получил ученую степень в области информационных технологий.

## Часть I

# Основные сведения о jQuery

В этой части книги вы познакомитесь с историей появления технологии jQuery и ее базовыми возможностями. Вы получите общее представление о концепциях jQuery, а после изучения материала по объектно-ориентированным средствам PHP, изложенного в части II, сможете выполнить упражнения, приведенные в части III (где разрабатывается проект реального приложения с использованием jQuery и PHP).



## Глава 1

# Введение в технологию jQuery

Чтобы объяснить роль технологии jQuery и ее приложений в современном веб-программировании, целесообразно немного остановиться на том, как она возникла, зачем создавалась и что представляло собой программирование на JavaScript до появления jQuery.

В этой главе вы узнаете о том, что такое библиотеки JavaScript, какие задачи они должны решать и почему библиотека jQuery завоевала признание большинства веб-разработчиков. Вы также изучите основы jQuery, разобравшись с тем, как сделать эту библиотеку доступной для своих приложений и как работает ее ядро — мощный селекторный движок.

## К вопросу о выборе jQuery

JavaScript считается довольно неудобным языком для написания веб-приложений. Отсутствие унифицированной поддержки различными браузерами, трудности отладки и пугающий синтаксис способны лишить новичков веры в возможность успешного освоения этого языка.

Справедливости ради следует отметить, что элементы, создающие видимые трудности, в то же время придают мощь этому языку, хотя вряд ли данный аргумент сделает JavaScript более привлекательным для начинающих веб-разработчиков и побудит их добавить этот язык в свой рабочий арсенал.

## Библиотеки JavaScript

Процесс изучения JavaScript всегда был болезненным для разработчиков, и по мере того, как всеобщее чувство недовольства нарастало, некоторые из них взялись создавать **библиотеки** JavaScript, которые часто называют **JavaScript-фреймворками**.

Библиотеки этого типа призваны упростить использование языка JavaScript и сделать его более доступным как для новичков, так и для разработчиков со стажем за счет предоставления простых в применении функций, облегчающих решение повседневных задач. Эффективность библиотек становится особенно заметной при работе с *асинхронным JavaScript и XML (AJAX)*, поскольку решение соответствующих задач с применением непосредственно JavaScript оказывается более сложным.

Библиотеки JavaScript предоставляют упрощенный синтаксис для решения типовых задач, что позволяет сократить сроки разработки и ускорить процесс обучения новичков. Кроме того, они частично упрощают написание сценариев JavaScript, не зависящих от типа браузера, беря на себя всю работу по проверке совместимости

с помощью собственных встроенных средств, что обеспечивает *огромную* экономию времени на этапе написания программного кода.

---

**Примечание.** Подробнее о различиях в использовании AJAX и непосредственно JavaScript говорится в главе 2.

---

Существует довольно много библиотек JavaScript. Самыми популярными из тех, которые используются разработчиками в настоящее время, являются библиотеки Prototype (<http://www.prototypejs.org>), MooTools (<http://mootools.net>), Yahoo! UI Library (<http://developer.yahoo.com/ui>), а также главный предмет рассмотрения данной книги — jQuery.

## Преимущества, обеспечиваемые jQuery

Каждой библиотеке JavaScript свойственны свои преимущества. Не составляет исключения и библиотека jQuery, которая обладает следующими достоинствами:

- небольшой размер файла (около 23 Кбайт для версии 1.4);
- чрезвычайно простой синтаксис;
- возможность объединения последовательно вызываемых методов в цепочки;
- простая архитектура подключаемых модулей, расширяющих базовые возможности фреймворка;
- огромное сетевое сообщество пользователей;
- великолепная документация, доступная по адресу <http://api.jquery.com>;
- полезные расширения, такие как jQuery UI, предоставляющие дополнительную функциональность.

## История возникновения jQuery

Библиотека jQuery, детище Джона Резига, была впервые представлена общественности на компьютерной конференции BarCamp в Нью-Йорке в начале 2006 года (более подробную информацию о международной сети конференций BarCamp см. на сайте <http://barcamp.org>). Как отметил на своем сайте сам Резиг, к созданию jQuery его подтолкнули неудовлетворенность существовавшими на то время библиотеками и осознание того факта, что их можно значительно улучшить, уменьшив объем “синтаксической шелухи” и введя специфические элементы управления для часто выполняемых операций (<http://ejohn.org/blog/selectors-in-javascript/>).

Библиотека jQuery сразу же завоевала признание сообщества разработчиков и быстро распространилась среди его членов. В процесс усовершенствования библиотеки включились другие разработчики, итогом чего стал выпуск первой устойчивой версии 1.0, состоявшийся 26 августа 2006 года.

С тех пор библиотека jQuery непрерывно улучшается (к моменту написания данной книги была выпущена версия jQuery 1.4.2), и в настоящее время она снабжена множеством дополнений (подключаемых модулей), предложенных сообществом разработчиков. **Подключаемый модуль**, или **плагин** (plug-in), — это расширение, не входящее в состав основной библиотеки. Подробнее о подключаемых модулях jQuery вы узнаете в главе 10.

## Настройка тестовой среды

Поскольку лучший способ освоения нового языка — практические занятия, вам понадобится тестовая среда, предназначенная для выполнения учебных упраж-



нений с использованием jQuery. К счастью, настройка этой тестовой среды представляет собой простой двухшаговый процесс, включающий установку Firefox и Firebug.

На протяжении всей книги будет предполагаться, что вы используете браузер Firefox с подключаемым модулем Firebug, поскольку они предоставляют в ваше распоряжение превосходную тестовую консоль JavaScript.

## Установка Firefox

Чтобы установить на своем компьютере Firefox, посетите сайт <http://firefox.com> и запустите последнюю версию Firefox (версия 3.6 на момент написания книги). После завершения программы установки (Firefox Setup x.x.x для Windows или Firefox x.x.x для Mac) можно запустить Firefox.

## Установка Firebug

Чтобы установить Firebug, посетите, используя браузер Firefox, сайт <http://getfirebug.com> и щелкните на кнопке Install Firebug For Firefox для открытия диалогового окна программы установки (рис. 1.1). Щелкните на кнопке Установить сейчас (Install Now) и дождитесь окончания процесса установки. После этого перезапустите Firefox.

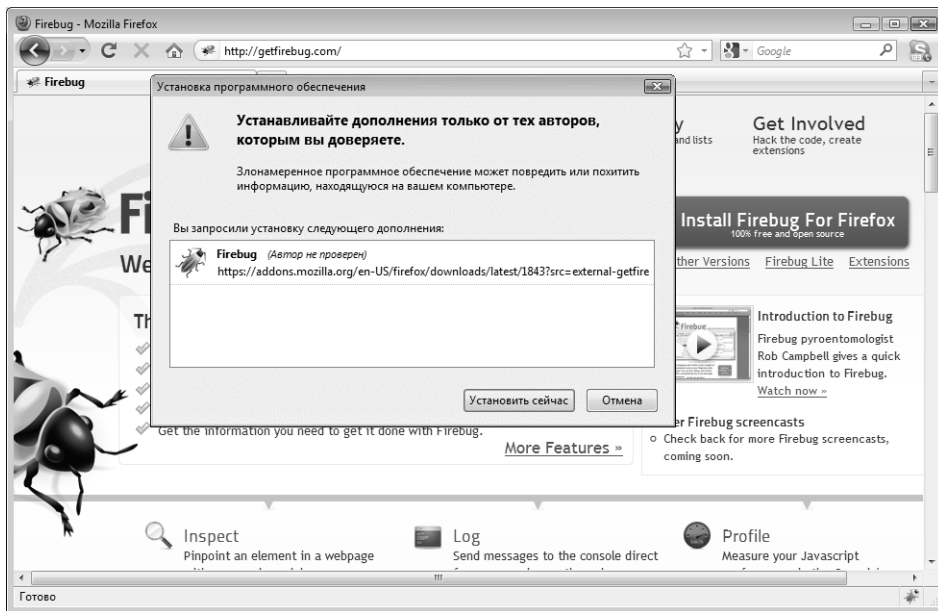


Рис. 1.1. Диалоговое окно программы установки Firebug

Перезапустив Firefox, вы увидите, что в правой части строки состояния появился новый значок с изображением жучка. Щелчок на этом значке выведет на экран строку элементов управления Firebug, начинающуюся с элемента Консоль (Console) (рис. 1.2).



Рис. 1.2. Так выглядит открытая панель консоли Firebug

**Примечание.** Спектр полезных функций Firebug не ограничивается отладкой JavaScript. Это дополнение окажет неоценимую помощь любому веб-разработчику. Более подробные сведения о Firebug можно получить по адресу <http://getfirebug.com>.

### Настройка локальной тестовой среды

Настраивать локальную тестовую среду, позволяющую выполнять приведенные в книге учебные упражнения, вовсе не обязательно, но если вы это сделаете, то поступите весьма мудро. Локальное тестирование делает процесс разработки более быстрым и безопасным и обычно выполняется проще, чем в случае использования удаленного сервера для этих целей.

#### Установка XAMPP

Чтобы быстро и без особых усилий развернуть локальную среду разработки на своем компьютере, загрузите и установите сервер XAMPP<sup>1</sup>, придерживаясь следующей процедуры.

1. Посетите сайт <http://www.apachefriends.org/en/xamp.html> и загрузите самую последнюю версию XAMPP, соответствующую вашей операционной системе.
2. Откройте загруженный файл. На компьютере Windows запустите EXE-файл, выберите папку и установите пакет. На компьютере Mac смонтируйте файл DMG и перетащите папку `xampp` в свою папку Applications.
3. Откройте панель управления XAMPP (XAMPP Control Panel), запустив файл `xampp-control.exe`, находящийся в папке `xampp`, и запустите сервер Apache.
4. Дабы удостовериться в том, что XAMPP работает, перейдите в браузере на страницу `http://localhost/`. Если сервер функционирует нормально, на экране отобразится домашняя страница XAMPP.

<sup>1</sup> Легко устанавливаемая кроссплатформенная сборка веб-сервера, включающая в себя собственно веб-сервер Apache, СУБД MySQL, PHP, Perl, FTP-сервер, диспетчерскую программу phpMyAdmin и многое другое. — *Примеч. ред.*

Помимо версий XAMPP для Windows и Mac существуют также дистрибутивные пакеты для операционных систем Linux и Solaris. Установка XAMPP в каждой из перечисленных операционных систем имеет свои особенности. Когда будете устанавливать и запускать локальную тестовую среду на своем компьютере, обязательно обратитесь к справочному руководству для получения дополнительной информации.

## Подключение jQuery к веб-страницам

Чтобы jQuery можно было использовать в проекте, библиотека должна быть загружена в HTML-документ, что позволит сценариям обращаться к библиотечным методам. Если библиотека не была предварительно загружена, то наличие любого сценария, в котором используется синтаксис jQuery, вероятнее всего, приведет к появлению ошибок JavaScript. К счастью, загрузить jQuery не составляет труда, и для этого существует ряд возможностей, которыми могут воспользоваться разработчики.

### Подключение загруженной копии библиотеки jQuery

Один из способов организации доступа к библиотеке jQuery состоит в том, чтобы сохранить ее копию в файловой структуре проекта и подключить ее, как любой другой файл JavaScript.

```
<script type="text/javascript" src="js/jquery-1.4.2.min.js">
/script>
```

### Подключение копии библиотеки jQuery, хранящейся на удаленном хосте

Еще один возможный вариант — включение в проект копии библиотеки jQuery, хранящейся на сайте Google Code. Это делается в надежде на то, что к моменту посещения вашего сайта пользователи уже будут иметь копию библиотеки, кешированную с какого-либо другого веб-сайта, который в свое время загрузил эту библиотеку с Google Code, что обеспечит сокращение времени загрузки страниц для пользователей вашего сайта.

Копия библиотеки, хранящаяся на удаленном сервере, включается в проект аналогично загруженной копии.

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js">
</script>
```

### Использование Google Libraries API

На сайте Google Code предлагается еще один вариант загрузки jQuery — с помощью Libraries API (<http://code.google.com/apis/ajaxlibs>). В соответствии с определением, которое дает Google, “Libraries API представляет собой сеть распространения содержимого и архитектуру для загрузки большинства популярных библиотек JavaScript с открытым исходным кодом”.

Использование Libraries API не вызывает никаких трудностей, и именно этот метод мы будем применять на протяжении всей книги. Чтобы подключить jQuery к своему веб-сайту с помощью Libraries API, используйте следующий фрагмент кода.

```
<script type="text/javascript"
src="http://www.google.com/jsapi"></script>
```

```
<script type="text/javascript">
  google.load("jquery", "1.4.2");
</script>
```

## Создание тестового файла

Настроив тестовую среду, создайте в папке `htdocs` установленной на вашем компьютере сборки XAMPP подпапку `testing`, а в ней — новый файл с именем `index.html`. Выбрав любой удобный для вас текстовый редактор, введите в этот файл показанный ниже HTML-код.

```
<!DOCTYPE html>
<html>
<head>
  <title>Тестирование jQuery</title>
</head>
<body>
  <p>Привет всем!</p>
  <p class="foo">Другой абзац, но уже с классом.</p>
  <p><span>Это контейнер SPAN внутри абзаца.</span></p>
  <p id="bar">Абзац с идентификатором.
    <span class="foo">А это контейнер SPAN внутри него.</span>
  </p>

  <script type="text/javascript"
    src="http://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("jquery", "1.4.2");
  </script>
</body>
</html>
```

---

**Примечание.** Чтобы избежать блокирования загрузки других элементов страницы, например изображений, вставляйте код, загружающий библиотеку, непосредственно перед закрывающим дескриптором тела документа (`</body>`). Тем самым вы также предотвратите выполнение сценария JavaScript до полной загрузки всех элементов страницы, что могло бы привести к непредвиденному поведению программы или появлению ошибок JavaScript.

---

Сохраните файл и выполните в Firefox переход по адресу: `http://localhost/testing` (рис. 1.3).

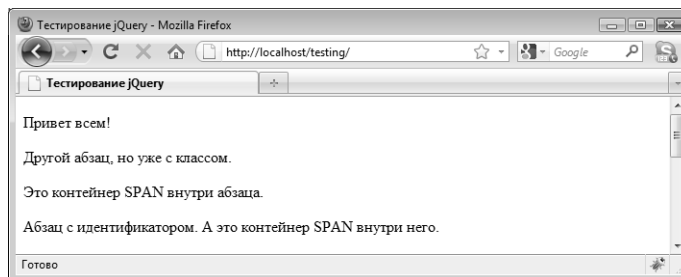


Рис. 1.3. Тестовый файл, загруженный в Firefox