

Предисловие

Приступая к написанию этой книги в начале 2008 года, я видел в этом насущную необходимость, поскольку в имевшейся тогда литературе отсутствовало подробное изложение самых важных составляющих языка программирования JavaScript (функций, замыканий и прототипов), а также приемов написания кросс-браузерного кода. К сожалению, ситуация с тех пор, как ни странно, не стала лучше.

Все больше сил и энергии вкладывается в разработку новых технологий, в том числе берущих свое начало от стандарта HTML5 или новых версий ECMAScript. Но нет никакого смысла осваивать новые технологии или применять самые современные библиотеки, не имея надлежащего представления об основных характеристиках языка JavaScript. Несмотря на самое светлое будущее разработки браузеров, главная задача в настоящее время — обеспечить работоспособность прикладного кода в большинстве браузеров и для большей части потенциальных пользователей.

Несмотря на то что эта книга писалась долго, она, к счастью, не утратила своей актуальности. Мой соавтор, Беэр Бибо, внес в нее существенные коррективы. Он постарался сделать так, чтобы материал книги еще долго оставался актуальным.

Долгое написание этой книги объясняется тем, что для материала последних ее глав, посвященных разработке кросс-браузерного кода, мне необходимо было приобрести известный опыт. Мои представления о разработке кросс-браузерного кода на практике опираются в основном на мою работу над библиотекой jQuery для JavaScript. Работая над материалом последних глав, посвященных разработке кросс-браузерного кода, я осознал, что большую часть базового кода библиотеки jQuery можно было бы написать иначе, оптимизировать и сделать ее способной управлять большим числом браузеров.

Вероятно, самые значительные изменения, внесенные в jQuery вследствие написания этой книги, связаны с полным пересмотром базового кода этой библиотеки: от организации пассивного прослушивания сети на уровне отдельных браузеров до обнаружения доступных средств. Благодаря этому применение библиотеки jQuery стало практически неограниченным, исключая необходимость учитывать, что в браузерах всегда будут присутствовать характерные программные ошибки или отсутствовать отдельные средства.

В результате этих изменений в jQuery предусмотрены многие усовершенствования, произведенные в браузерах за два прошедших года: выпуск браузера Chrome компанией Google; широкое распространение пользовательских агентов по мере роста популярности мобильных вычислений; острое соперничество среди компаний Mozilla, Google и Apple за повышение производительности их браузеров; а также значительные усовершенствования браузера Internet Explorer, на которые, наконец-то, решилась корпорация Microsoft. Теперь уже нельзя допускать, что один и тот же механизм визуализации (например, WebKit или Trident в Internet Explorer) будет всегда действовать одинаково. Коренные перемены происходят довольно быстро и распространяются среди постоянно растущего числа пользователей.

С помощью способов и приемов, описываемых в этой книге, кросс-браузерные возможности jQuery обеспечивают довольно прочную гарантию того, что код, написанный средствами jQuery, будет работать в наибольшем числе сред браузеров. Благодаря этой гарантии библиотека jQuery нашла широкое распространение за последние четыре года. По данным, опубликованным на веб-сайте BuiltWith.com, она теперь применяется на 58% из 10 тысяч самых широко посещаемых в Интернете веб-сайтов.

Средства JavaScript, в том числе вычисление кода, противоречивые операторы `with` и таймеры, относительно постоянны и продолжают использоваться самыми интересными способами. В настоящее время на основе JavaScript построен или скомпилирован целый ряд активно применяемых языков программирования, в том числе CoffeeScript и Processing.js. Но для того чтобы они действовали эффективно, требуется сложный синтаксический анализ языковых конструкций, вычисление кода и манипулирование областью действия. Несмотря на то что динамическое вычисление кода пользуется недоброй славой в силу его сложности и потенциальных нарушений безопасности, без него стало бы просто невозможным появление языка программирования CoffeeScript, который, в свою очередь, повлиял на составление спецификации языка ECMAScript.

Я лично пользуюсь всеми этими средствами до сих пор в своей работе в Академии Хана. Динамическое вычисление кода в браузере является весьма эффективным средством, позволяя создавать встраиваемые в браузеры среды программирования и внедрять такие необычные приемы, как внесение кода во время фактического выполнения. Все это возбуждает дополнительный интерес к изучению программирования на компьютере и открывает новые возможности, недоступные для традиционных средств обучения.

Будущее разработки браузеров остается весьма прочным и, главным образом, благодаря средствам, встроенным в JavaScript, и прикладным интерфейсам API самих браузеров. Имея основательное представление о самых важных составляющих языка JavaScript, а также желание писать код, способный работать во многих браузерах, вы будете в состоянии разрабатывать изящный, быстродействующий и повсеместно применяемый код.

ДЖОН РЕЗИГ

Благодарности

Количество людей, принявших участие в работе над этой книгой, способно многих удивить. Свой плодотворный вклад в книгу, которую вы держите в руках или читаете на экране в электронном виде, внесли многие одаренные люди.

Сотрудники издательства Manning Publications неутомимо потрудились, чтобы добиться такого качества издания, на которое надеялись авторы книги, за что мы им искренне благодарны. Ведь без них эта книга не увидела бы свет. Мы выражаем благодарность не только издателю Маржан Бэйс (Marjan Vasce) и главному редактору Майку Стивенсу (Mike Stephens), но и следующим сотрудникам издательства: Джеффу Блейлу (Jeff Bleiel), Дугласу Паднику (Douglas Pudnick), Себастьяну Стирлингу (Sebastian Stirling), Андреа Каухер (Andrea Kaucher), Карен Тегтмайер (Karen Tegtmayer), Кэйти Теннат (Katie Tennant), Меган Йоки (Megan Yockey), Дотти Марсико (Dottie Marsico), Мэри Пирджиз (Mary Piergies), Энди Кэроллу (Andy Carroll), Мелоди Долаб (Melody Dolab), Тиффани Тейлор (Tiffany Taylor), Деннису Далиннику (Dennis Dalinnik), Гэбриэлу Добреску (Gabriel Dobrescu), а также Рону Томичу (Ron Tomich).

Невозможно выразить словами нашу признательность коллегам, рецензировавшим книгу и помогавшим нам довести ее до состояния полной готовности к печати: от вылавливания простых опечаток до исправления ошибок в терминах и коде и вплоть до организации книги по отдельным главам. Каждый из них внимательно просмотрел рукопись книги, прежде чем она была сдана в печать. За этот нелегкий труд мы хотели бы поблагодарить Алессандро Галло (Alessandro Gallo), Андре Робержа (Andr y Roberge), Остина Кинга (Austin King), Остина Циглера (Austin Ziegler) Чэда Дэвиса (Chad Davis), Чарльза Е. Логстона (Charles E. Logston), Криса Грэя (Chris Gray), Кристофера Хаупта (Christopher Haupt), Крэйга Ланкастера (Craig Lancaster), Кэртиса Миллера (Curtis Miller), Даниэля Бретуа (Daniel Bretoi), Дэвида Веддера (David Vedder), Эрика Арвидссона (Erik Arvidsson), Гленна Стокола (Glenn Stokol), Грэга Дональда (Greg Donald) Джеймса Хэтуэя (James Hatheway), Джаред Хирша (Jared Hirsch), Джима Руза (Jim Roos), Джо Литтона (Joe Litton), Йохана Линка (Johannes Link), Джона Польсона (John Paulson), Джошуа Хейера (Joshua Heyer), Джулио Гвиджарро (Julio Guijarro), Курта Янга (Kurt Jung), Лоика Саймона (Lo c Simon), Нила Микса (Neil Mix), Роберта Хансона (Robert Hanson), Скотта Соьера (Scott Sauyet), Стюарта Каборна (Stuart Caborn), а также Тони Ниманна (Tony Niemann).

Особая благодарность выражается Валентину Креттазу (Valentin Crettaz), научному редактору книги. Помимо проверки каждого примера кода в нескольких средах, он внес неоценимый вклад в уточнение текста рукописи и обнаружение пропущенной информации с учетом последних изменений, внесенных в поддержку JavaScript и HTML5 в браузерах. Не меньшая благодарность выражается также Берту Бэйтсу (Bert Bates), давшему отзыв о книге и неоценимые предложения по поводу ее улучшения. Бесконечные часы общения с ним по Skype, безусловно, окупились сторицей.

Джон Резиг

Мне хотелось бы поблагодарить моих родителей за их постоянную моральную и материальную поддержку в течение многих лет. Они предоставили все необходимые ресурсы и средства, чтобы пробудить во мне интерес к программированию, и с тех пор постоянно поощряли мои занятия в данной области.

Безр Бибо

За участие в работе над этой, уже пятой для меня книгой я хотел бы, как всегда, поблагодарить своих коллег, участников и организаторов форума JavaRanch (www.javaranch.com). Ведь без личного участия в этом форуме я бы никогда не начал писать книги, и поэтому я искренне благодарю Пола Уитона (Paul Wheaton) и Кэти Сьерра (Kathy Sierra) за добрые напутствия, а также моральную поддержку следующих коллег, хотя и не только их одних: Эрика Паскарелло (Eric Pascarello), Эрнста Фридмана Хилла (Ernest Friedman Hill), Эндрю Монкхауза (Andrew Monkhouse), Джина Боярски (Jeanne Boyarsky), Берта Бэйтса (Bert Bates), а также Макса Хабиби (Max Habibi).

Сердечная признательность выражается моему другу Джею и моим собакам, Медвежонку и Козмо, за то, что они терпели мое присутствие и редко наделялись должным вниманием с моей стороны во время работы над этой книгой. И наконец, мне хотелось бы поблагодарить своего соавтора Джона Резига, без которого эта книга не состоялась бы.

Об этой книге

Особое значение JavaScript было очевидно не всегда, но теперь оно несомненно. Веб-приложения служат для того, чтобы предоставить пользователям функционально богатый и удобный интерфейс, но без JavaScript в Интернете можно лишь показывать фотографии кошек. Теперь, как никогда прежде, разработчикам веб-приложений требуются прочные знания и навыки программирования на языке JavaScript, приводящем в действие эти приложения.

Но как и завтрак с апельсиновым соком, язык JavaScript полезен не только для браузеров. Преодолев узкие границы применения в браузерах, этот язык программирования теперь применяется на серверах в таких механизмах, как Rhino и V8, а также в интегрированных средах вроде Node.js. И хотя эта книга посвящена главным образом применению JavaScript в веб-приложениях, основы этого языка программирования, представленные в части II, выходят далеко за пределы применимости при разработке веб-приложений. С увеличением числа разработчиков, пользующихся JavaScript, теперь, как никогда прежде, стало очень важно знать твердо основы этого языка программирования, чтобы владеть им в совершенстве.

Кому адресована книга

Эта книга рассчитана на тех, кто уже знаком с JavaScript. Если же вы только начинаете изучать язык JavaScript или знаете лишь дюжину его операторов из фрагментов кода, найденных в Интернете, значит, эта книга пока еще не для вас. Книга адресована разработчикам веб-приложений, уже владеющим основами программирования на JavaScript. А это означает, что вы должны знать основную структуру операторов JavaScript и уметь составлять из них простые страничные сценарии. Для этого совсем не обязательно иметь большой опыт программирования на JavaScript. Этот опыт можно приобрести,

проработав материал данной книги. Но и полным невеждой в JavaScript также нельзя быть.

Для чтения этой книги требуются также знания HTML и CSS. Опять же эти знания и опыт совсем не обязательно должны быть основательными, тем не менее вы должны владеть основами компоновки веб-страниц. Если же вам требуется материал для предварительной подготовки к чтению этой книги, рекомендуем приобрести одно из следующих популярных изданий по JavaScript и разработке веб-приложений: *JavaScript: The Definitive Guide* Дэвида Флэнагана (David Flanagan; в русском переводе книга вышла под названием *JavaScript. Подробное руководство*, *JavaScript: The Good Parts* Дугласа Крокфорда (Douglas Crockford; в русском переводе книга вышла под названием *JavaScript. Сильные стороны*) или *Head First JavaScript* Майкла Моррисона (Michael Morrison; в русском переводе книга вышла под названием *Изучаем JavaScript*)

Структура книги

Эта книга организована так, чтобы вы смогли пройти весь курс обучения от ученика до мастера программирования на JavaScript. В части I представлены основной предмет книги и ряд инструментальных средств для изучения материала остальных частей. В части II главное внимание уделяется основам JavaScript, т.е. тем языковым средствам, которые воспринимаются как вполне естественные, но еще нужно знать, каким образом они действуют. Возможно, это самая важная часть книги, и даже если вы осилите только ее, то и тогда получите намного более основательное представление о JavaScript. В части III основы, представленные в части II, рассматриваются более углубленно в связи с решением трудных задач, которые перед нами ставят браузеры. И в завершающей книге части IV расширенные языковые возможности рассматриваются на основе уроков, извлеченных из создания таких усовершенствованных библиотек JavaScript, как jQuery.

А теперь перейдем к краткому изложению отдельных глав. В главе 1 представлены те трудности, которые приходится преодолевать разработчикам современных веб-приложений. В ней рассматриваются основные трудности, которые влечет за собой широкое распространение браузеров, а также апробированные передовые методики, которым необходимо следовать при разработке веб-приложений, включая тестирование и анализ производительности.

В главе 2 обсуждаются вопросы тестирования, а также текущее состояние процедур и инструментальных средств тестирования. В ней представлен также небольшой, но эффективный принцип утверждения, который будет в исключительном порядке применяться в остальной части книги с целью убедиться в том, что код выполняет свои функции, а иногда и в том, что он их не выполняет!

Вооружившись упомянутыми выше инструментальными средствами, можно перейти к главе 3, с которой начинается рассмотрение основ языка и, в частности, с понятия *функции* в том виде, в каком оно определяется в JavaScript. На первый взгляд, основное внимание следовало бы уделить понятию *объекта*, тем не менее нужно иметь прежде всего ясное представление о функции, поскольку JavaScript — язык функционального программирования. Ведь именно с понятия функции начинается наше постепенное превращение из обычных учеников в мастеров программирования на JavaScript!

Не завершив до конца рассмотрение функций, основы, усвоенные в главе 3, можно применить уже в главе 4 для решения задач, возникающих при разработке веб-приложений. В этой главе понятие рекурсии рассматривается не только ради нее са-

мой, но и для более углубленного изучения функций при тщательном исследовании принципа ее действия. Здесь же поясняется, каким образом аспекты функционального программирования можно применять для получения изящного, надежного и лаконичного кода и как обращаться со списками, содержащими переменное число аргументов, а также перегружать функции в языке, где отсутствует собственная поддержка объектно-ориентированного понятия перегрузки методов.

В главе 5 рассматриваются замыкания — одно из самых важных понятий не только в этой книге, но и в функциональном программировании вообще. Замыкания предоставляют мелкоструктурный контроль над областью действия объектов, объявляемых и создаваемых в прикладных программах. Контроль над этими областями действия имеет решающее значение для искусного написания кода. И даже если вы оставите чтение книги на главе 5, хотя мы надеемся, что вы этого все же не сделаете, то и тогда ваша квалификация в разработке программ на JavaScript станет намного выше, чем перед началом чтения книги.

Объекты, наконец-то, представлены в главе 6, где поясняется создание шаблонов объектов с помощью свойства `prototype` функции, а также связывание объектов с функциями для их определения. Это одна из причин, по которым сначала рассматриваются функции, а затем объекты.

В главе 7 основное внимание уделяется регулярным выражениям — языковым средствам, которые нередко упускаются из виду, но могут заменить собой немало строк кода, если они используются правильно. В этой главе будет показано, каким образом строятся и применяются регулярные выражения и насколько изящно некоторые типичные затруднения разрешаются с помощью регулярных выражений и методов, в которых они поддерживаются.

Часть II, посвященная основам JavaScript, завершается главой 8, где рассматривается использование таймеров и интервалов в модели однопоточной обработки, применяемой в JavaScript. С помощью *рабочих веб-процессов* в HTML5 предполагается преодолеть ограничения, присущие однопоточной обработке, но большинство браузеров пока еще для этого не приспособлены, и поэтому практически весь существующий код JavaScript зависит от ясного представления о модели однопоточной обработки, применяемой в JavaScript.

Часть III начинается с главы 9, в которой раскрывается черный ящик, содержащий механизм вычисления кода JavaScript во время выполнения. В этой главе рассматриваются различные способы оперативного вычисления кода, включая соблюдение безопасности и выбранной области действия. На реальных примерах показывается реализация вычислений кода в формате JSON, метаязыков (т.е. предметно-ориентированных языков программирования), сжатия и запутывания кода и даже элементов аспектно-ориентированного программирования.

В главе 10 рассматривается противоречивый оператор `with`, служащий для укорочения ссылок в области действия. Как бы ни относиться в оператору `with` (с восторгом или полным отвращением), он присутствует в большей части рабочего кода, и поэтому нужно иметь ясное представление о нем.

Вопросы написания кросс-браузерного кода рассматриваются в главе 11. В связи с этими вопросами в ней по порядку важности обсуждаются пять основных проблем разработки веб-приложений: отличия браузеров, программные ошибки и их устранение, внешний код и разметка документов, отсутствующие средства и регрессии. Для оказания помощи в разрешении кросс-браузерных затруднений подробно рассматриваются такие методики, как имитация средств и обнаружение объектов.

Обращению с атрибутами элементов разметки, свойствами и стилями уделяется основное внимание в главе 12. Со временем отличия в обращении с подобными характеристиками элементов разметки в браузерах сотрутся, а до тех пор существует целый ряд сложных затруднений, разрешению которых посвящена эта глава.

В главе 13, завершающей часть III, подробно рассматриваются особенности обработки событий в браузерах и способы построения единой подсистемы обработки событий независимо от конкретного браузера. В такую подсистему входят средства, не предоставляемые браузерами, в том числе специальные события и делегирование событий.

В части IV происходит плавный переход к подробному рассмотрению расширенных возможностей, извлекаемых из сердцевины библиотек JavaScript типа jQuery. Так, в главе 14 показывается, каким образом строятся программные интерфейсы API для манипулирования объектной моделью документов (DOM) во время выполнения, включая прерывание гордиева узла внесения новых элементов в модель DOM.

И наконец, в главе 15 обсуждаются вопросы построения механизмов CSS-селекторов и различные способы синтаксического анализа и вычисления селекторов в подобных механизмах. Эта глава не для малодушных, но ее стоит прочитать, чтобы проверить, насколько вы овладели искусством программирования на JavaScript.

Условные обозначения, принятые в книге

Весь исходный код в листингах или тексте книги набран **моноширинным** шрифтом, чтобы отделить его от обычного текста. Имена функций, методов и элементов разметки в коде XML и их атрибутов также набраны **моноширинным** шрифтом.

В некоторых случаях исходный код переформатирован для того, чтобы он уместился на страницах книги. Как правило, исходный код написан с учетом ограничений по ширине страниц книги, но иногда могут встречаться незначительные отличия в его форматировании в книге по сравнению с загружаемым вариантом исходного кода. И лишь в некоторых случаях, когда длинные строки кода нельзя переформатировать, не изменяя его смысл, в листингах на страницах книги указываются метки продолжения строк кода.

Многие листинги снабжены комментариями к коду, поясняющими наиболее важные понятия. Но, как правило, нумерованные метки указывают на пояснения в следующем далее тексте.

Загружаемый исходный код

Исходный код для проработки примеров, приведенных в этой книге (наряду с некоторыми дополнениями, отсутствующими в тексте книги), свободно доступен для загрузки на посвященной книге веб-странице по адресу www.manning.com/SecretsoftheJavaScriptNinja. Исходный код примеров организован в папках по отдельным главам. Такая организация специально подготовлена для размещения на локальном веб-сервере, например Apache HTTP Server. Для этого достаточно извлечь загруженный код из архива в избранную папку и сделать ее корневой для документов веб-приложения.

За некоторыми исключениями, для большинства примеров наличие веб-сервера вообще не требуется, и при желании их исходный код можно загрузить на выполнение непосредственно в браузер. Все примеры были проверены в различных браузерах,

существовавших на момент написания книги, в том числе Internet Explorer 9, Firefox, Safari и Google Chrome.

Как связаться с авторами

Авторы и издательство Manning Publications приглашают читателей на форум, посвященный этой книге, где они могут разместить свои комментарии к книге, задать вопросы технического характера и получить помощь от авторов и других пользователей JavaScript. Для доступа и подписки на форум введите в окне своего браузера адрес www.manning.com/SecretsoftheJavaScriptNinja и щелкните на ссылке Author Online (Автор в оперативном режиме). На открывшейся странице появятся сведения о том, как войти на форум после регистрации, о видах доступной помощи, а также правилах ведения форума.

Об иллюстрации на обложке книги

Фигура воина на обложке книги взята с гравюры на дереве “Актер *но* в роли самурая” неизвестного японского художника середины XIX века. *Но* — это форма музыкальной драмы в классическом японском театре, исполняющейся начиная с XIV века. Многие персонажи подобных драм играют в масках, причем мужские и женские роли исполняют только мужчины. Самурай — это героическая фигура японской истории, нередко представленная в театре и изобразительном искусстве с большим мастерством, пышностью наряда и суровостью настоящего воина.

Самурай и ниндзя были воинами, отличавшимися в японском военном искусстве своим мастерством, смелостью и хитростью. Самурай относились к военной элите, были хорошо образованы, умели читать и писать, а также искусно воевать. Они были связаны строгим кодексом чести под названием бусидо, что означает путь воина. Этот кодекс чести передавался изустно из поколения в поколение, начиная с X века. Самураев набирали из аристократов и высших слоев японского общества аналогично европейским рыцарям. Они шли на битву в строгом боевом порядке в сложных доспехах и ярких одеждах, чтобы произвести впечатление и устрашить противника. А ниндзя отличались в большей степени своим боевым искусством, чем общественным положением или образованностью. Одевались они в черное, закрывая свое лицо, выполняли боевые задания в одиночку или небольшими группами, нападая на противника скрытно, с хитрыми уловками и применяя любую тактику для достижения успеха. Единственным правилом их поведения была скрытность.

Иллюстрация на обложке книги отобрана с трех японских гравюр, которые многие годы принадлежат главному редактору издательства Manning Publications. И когда авторы искали подходящее изображение ниндзя для обложки этой книги, их внимание привлекла поразительная гравюра самурая, которую они отобрали в качестве иллюстрации благодаря ее сложной детализации, ярким краскам и выразительному изображению сурового воина, готового нанести удар и победить.

В наше время, когда одну компьютерную книгу трудно отличить от другой, издательство Manning Publications славится своей изобретательностью и инициативностью в оформлении обложек издаваемых более двадцати лет книг, где наглядно проявляется все многообразие мирового изобразительного искусства. И характерным тому примером служит обложка этой книги.