

*Посвящается Весте, богине домашнего очага
и хранительнице всех программистов*

Предисловие

В программировании важна вежливость.

Разработка Web-приложений чаще всего сводится к поиску путей преодоления проблем при решении тривиальных задач. Вероятно, основной и самый неприятный источник этих проблем — браузер. Помимо ошибок, имеющихся во всех основных браузерах, и несоответствия этих браузеров спецификациям, этим браузерам присущи определенные ограничения — недоступные элементы управления, специфика выполнения кода на JavaScript и еще множество неприятных вещей.

Постарайтесь забыть об этом. В этой книге мы рассмотрим библиотеки Prototype и Scriptaculous, созданные для того, чтобы позволить вам сосредоточиться на действительно важной вещи: воплощении в жизнь ваших идей.

Основная предпосылка достижения цели — вежливость. Эта концепция лежит в основе обеих рассматриваемых в книге библиотек. Во-первых, библиотеки вежливы к вам, как к разработчику. Чаще всего, чтобы решить задачу, вам достаточно написать несколько строк кода. Кроме того, в библиотеках используется единый стиль оформления кода, так что вам не придется долго разбираться в том, что делает каждая отдельно взятая функция. Исходный код оптимизирован для облегчения понимания — просто потратьте немного времени! В результате вы сможете создавать приложения с удобным пользовательским интерфейсом, достигая вежливости и по отношению к пользователю. Вы можете модифицировать элементы управления, чтобы создавать изощренные пользовательские интерфейсы, тратя при этом очень немного времени и не заботясь о том, как работают базовые элементы.

Библиотеки Prototype и Scriptaculous — не первые, в которых используется этот подход. Они родились в среде разработки Ruby on Rails, и если вы работаете в ней, то многие вещи, рассматриваемые в этой книге, будут вам знакомы. Если вы не работаете в Rails, попробуйте!

Вежливость проявляется во множестве разных вещей. Она относится ко всем аспектам разработки Web-сайтов — от создания серверных частей приложений до оформления пользовательского интерфейса. Как сказал один из разработчиков о библиотеке Scriptaculous, “эффект Puff заставил меня плакать от радости”. Хотя вы, возможно, и не столь эмоциональны, вы наверняка найдете в этих библиотеках много полезного. Возможно, самый важный результат их применения — это возможность вовремя возвращаться домой и при этом успевать делать свою работу.

Попробуйте создать еще один вежливый сайт в сети, поделитесь опытом с другими разработчиками или создайте дополнения и расширения к Prototype и Scriptaculous!

Томас Фукс,
СТО, wollzelle
Создатель Scriptaculous

Введение

Пару лет назад JavaScript считался языком для тех, кто не хочет заниматься настоящим программированием. Появление Ajax изменило отношение к этому языку, и теперь вы можете без стыда признаться, что программируете на JavaScript. Я сам прошел через это превращение и считаю, что благодаря этому достиг многого.

Я даже начал изучать основы самого языка и, к своему удивлению, обнаружил, что в нем есть особый стиль и логика, хотя он и не похож на Java. Я разобрался с цепочками прототипов и начал создавать объекты на JavaScript. Я был очень рад, когда разобрался в том, как работают в JavaScript закрытые фрагменты кода, и поздравлял себя с превращением в знатока языка. Как всегда, гордость предшествует падению.

В какой-то момент времени я узнал о существовании библиотеки Prototype и скопировал написанный кем-то код, использовавший ее. Этот код подставлял в набираемый текст слова, подбирая их по первым введенным буквам. Из любопытства я решил разобраться, как же этот код работает. Когда я впервые попытался просмотреть исходный код Prototype, он показался мне китайской грамотой — в лучшем случае чем-то похожим на Perl. Этот код совсем не был похож на те программы на JavaScript, которые писал я сам, и я решил, что осторожность — лучшая часть храбрости. Я убежал и спрятался.

Однако я не мог прятаться бесконечно и постепенно начал понимать общий стиль библиотеки Prototype. В то время я пытался изучить еще один язык программирования — Ruby, и я стал замечать сходство между ними. Постепенно я понял, что автор библиотеки знал о закрытых фрагментах кода, цепочках прототипов и обо всем остальном в JavaScript на порядок больше меня. Структура библиотеки была тщательно продумана, и благодаря этому несколько удачных строк кода можно было раз за разом применять в самых разнообразных ситуациях.

Библиотека Scriptaculous использует тот же подход, создавая сложные конструкции из проверенных небольших частей, и предоставляет доступ к этим конструкциям так, что новички могут их использовать, написав всего несколько строк кода. Читая лекции по Ajax, я знакомлю студентов с библиотекой Scriptaculous, и они неизменно удивляются, когда видят, как легко с ее помощью создавать замысловатые красочные интерфейсы для Web-приложений.

В этой книге как раз и рассказывается о том, как использовать эти библиотеки, написав для этого всего несколько строк кода. Prototype и Scriptaculous позволяют легко делать простые вещи и сосредотачиваться на решении основных задач, не отвлекаясь на вопросы кроссбраузерной совместимости или поддержки библиотек. С помощью этой книги вы можете разобраться и во внутренних механизмах этих библиотек, а также самого языка JavaScript.

Для меня библиотеки Prototype и Scriptaculous сделали программирование на JavaScript гораздо более удобным и интересным. Я надеюсь, это будет справедливо и для вас.

Дэйв Крейн

Благодарности

Мы хотели бы выразить благодарность множеству людей, участвовавших в претворении этого проекта в жизнь и вложивших в него свои идеи и труд.

Мы благодарим рецензентов, сообщавших нам свое мнение об этой книге в ходе ее написания. В их числе — Филип Хэлстром (Philip Hallstrom), Кристофер Гаупт (Christopher Haupt), Дерек Лэкин (Derek Lakin), Джон Тирсен (John Tirsen), Дипак Вохра (Deepak Vohra), Джефф Каннингем (Jeff Cunningham), Кристофер Бейли (Christopher Bailey), Скотт Шоу (Scott Shaw), Марк Игл (Mark Eagle) и Бенджамин Горлик (Benjamin Gorlick). Особая благодарность — Дипаку Вохре за правку книги перед ее отправкой в печать.

Мы очень благодарны читателям, которые участвовали в программе Manning Early Access Program и сообщали об ошибках, найденных в черновых вариантах книги.

Мы благодарим работников Manning Publications, особенно издателя Маржан Бэйс (Marjan Base) и наших редакторов Майка Стефенса (Mike Stephens) и Синтию Кейн (Cynthia Kane), а также издательский коллектив: Энди Кэрролла (Andy Carroll), Дотти Мариско (Dottie Marisco), Гордана Салиновича (Gordan Salinovic), Элизабет Мартин (Elisabeth Martin) и Мэри Пиржис (Mary Piergies).

Большое спасибо Томасу Фуксу (Thomas Fuchs) за то, что он согласился написать вступление к книге, и Тому Локи (Tom Locke) за его вклад в главу, посвященную Rails. Наша искренняя благодарность за ваш труд и участие.

Дэйв Крейн

Прежде всего я хотел бы поблагодарить Сэма Стефенсона (Sam Stephenson) и Томаса Фукса за создание этих замечательных библиотек, и команду, занимающуюся разработкой ядра Prototype, — Джастина Палмера (Justin Palmer), Эндрю Дюпонта (Andrew Dupont), Дэна Уэбба (Dan Webb), Скотта Реймонда (Scott Raymond), Мислава Маронича (Mislav Marohnic), Кристофа Портенью (Cristophe Porteneuve), Тоби Лэнгела (Tobie Langel) и Сета Диллингэма (Seth Dillingham) за быстроту и качество их работы.

Я также хочу поблагодарить моих коллег — Саймона Уоррика (Simon Warrick), Тима Уилсона (Tim Wilson), Сюзанну Эллис (Susannah Ellis), Саймона Кроссли (Simon Crossley), Роба Левайна (Rob Levine) и Майлза Уилсона (Miles Wilson) из Historic Futures за оказанную ими поддержку, а также Венди, Ника, Грэма, команду `Skillsmatter.com` и моих талантливых студентов — за помощь в воплощении моих замыслов в эту книгу.

Хочу выразить признательность Бэру и Тому. Благодаря их неоценимой помощи эта книга приобрела законченную форму.

И наконец, благодарю мою семью — Чиа, Бена и Софию — за способность выдерживать общение с человеком, пишущим очередную книгу, а также моих родителей за терпение и понимание.

Бэр Бибо

Я хочу поблагодарить моих друзей с сайта `javaran.ch.com`, которые убедили меня взяться за перо (скорее, за клавиатуру) и написать книгу, как только я заикнулся о возникновении такого желания. Среди этих друзей я хотел бы упомянуть Эрнеста Фридмана-Хилла (Ernest Friedman-Hill), Эрика Паскарелло (Eric Pascarello), Бена Саутера (Ben Souther), Макса Хабиби (Max Habibi), Марка Хершберга (Mark Herschberg) и Кэти Сиерры (Kathy Sierra).

Я хотел бы поблагодарить Пола Уитона, владельца `javaranch.com`, за создание замечательного сайта.

Я благодарю Дэйва Крейна и Майка Стефенса за их веру в меня и за то, что они позволили мне сделать свой вклад в их работу.

Я также благодарю моих собак, Гизмо и Литл Бэра, изображения которых встречаются на некоторых рисунках в этой книге без их согласия на это.

И наконец, спасибо моему другу Джею, который терпеливо выслушивал мою брань в адрес Word, браузеров и несовершенства Вселенной и который много лет назад познакомил меня с миром Мас.

Об этой книге

Prototype и Scriptaculous, как и многие другие программные библиотеки, предназначены для повышения эффективности работы программистов. Написание программ для Web-браузеров — своеобразная область программирования, более свободная и экспрессивная, чем работа с обычными десктопными приложениями, и одновременно лишенная некоторых базовых возможностей, доступных этим приложениям. Я знаю разработчиков десктопных приложений, пытавшихся создавать собственные библиотеки для поддержки перетаскивания, и нескольких разработчиков на DHTML/Ajax, пытавшихся сделать то же самое. Эта возможность нечасто является критичной для приложений — обычно она всего лишь инструмент для достижения цели. Тратить массу усилий на то, чтобы заставить инструмент работать, — это, в лучшем случае, не слишком умно. В худшем случае это может привести к провалу проекта.

Читатели

Эта книга предназначена для людей, программирующих на JavaScript и знающих, чего они хотят. Этим людям требуется возможность изящного и эффективного воплощения в жизнь своих идей или возможность создания программ с неординарными новыми возможностями. Библиотеки Prototype и Scriptaculous позволяют легко и быстро писать код, не тратя на это чересчур много времени. Если вы хотите возиться, объявляя собственную координатную систему, реализуя для нее все базовые механизмы, функции, алгоритмы и объекты, то эти библиотеки вам вряд ли пригодятся. Хотя, разумеется, вы можете воспользоваться ими и для этой цели...

Кроме того, эта книга адресована людям, обладающим определенным опытом в написании Ajax-приложений и желающим разобраться в том, как работают эти библиотеки. В книге мы рассмотрим большую часть компонентов этих библиотек и покажем, как их можно использовать. Возможно, еще важнее то, что мы покажем взаимосвязи между этими компонентами и общую структуру, которую они образуют.

Независимо от того, хотите ли вы просто написать эффективно работающую программу или улучшить свои навыки программирования на JavaScript, в этой книге вы найдете что-то полезное для себя.

Структура книги

Книга разделена на четыре части, каждая из которых посвящена определенной теме. В части I мы сосредоточимся на Web-клиентах, а точнее, на асинхронных запросах. Ajax предоставляет функциональность, которая радикально меняет структуру многих приложений, и Prototype предоставляет разнообразные механизмы для работы с Ajax.

В главе 1 кратко описаны библиотеки Prototype и Scriptaculous в целом и их применение в Ajax-приложениях. В главе 2 вы ознакомитесь с нашим основным приложением-примером, QuickGallery – сетевым просмотрщиком изображений. В главах 3 и 4 мы используем QuickGallery для экспериментов с различными элементами Ajax и их поддержкой в Prototype – от базовых компонентов, необходимых для работы с Ajax-запросами, до весьма замысловатых вспомогательных механизмов, появившихся в последних версиях Prototype.

В части II мы обратимся к библиотекам Scriptaculous – каждая из глав этой части посвящена какому-то из ключевых компонентов. В главе 5 мы рассмотрим подсистему Effects – начиная с однократного вызова нужного спецэффекта и заканчивая составлением собственных эффектов из сочетаний уже существующих.

Глава 6 посвящена подсистеме Controls, которая предоставляет готовые к использованию элементы управления, поддерживающие Ajax. Здесь мы тоже пройдем весь путь от простейших примеров использования элементов до их настройки и изучения механизмов работы. Кроме того, мы рассмотрим привязку этих элементов к процессам в серверных частях приложений и организацию интерактивных приложений.

В главе 7 мы рассмотрим поддержку перетаскивания (drag-and-drop) в библиотеке Scriptaculous. В этой библиотеке механизмы поддержки перетаскивания состоят из нескольких уровней – базовые возможности предоставляются относительно простыми объектами, и из этих объектов составляются сложные элементы пользовательских интерфейсов. Мы рассмотрим и устройство этих элементов, и их применение в программах.

Часть III предназначена в основном для тех, кто хочет углубить свои познания в JavaScript. Этот язык легко можно расширять, и библиотека Prototype добавляет в него очень полезные, хотя и необычные, структуры. Мы рассмотрим новые возможности, которые Prototype добавляет в базовые объекты языка, и проиллюстрируем каждую такую возможность маленьким работающим примером. Дабы максимально полно рассмотреть все новые возможности, мы приводим в этой части небольшие примеры, которые выполняются с помощью специальной программы.

В главе 8 мы рассмотрим класс Object – базовый класс, на котором основываются все другие классы объектов в JavaScript. Мы разберемся, как можно создавать новые объекты и многократно используемые типы объектов и как использовать основанную на прототипах систему наследования в JavaScript. Затем мы покажем, как Prototype упрощает использование этих механизмов, и научим вас создавать собственные иерархии объектов с помощью Prototype.

В главе 9 описаны функции JavaScript и связанная с ними концепция закрытых фрагментов кода. Это маленькая, но важная глава, в которой рассматривается один из самых мощных и одновременно самых трудных для понимания механизмов JavaScript. С закрытыми фрагментами кода трудно работать, но Prototype позволяет упростить этот процесс.

Глава 10 посвящена массивам JavaScript. Можно без преувеличения сказать, что Prototype совершенно изменяет способы работы с массивами, добавляя более тридцати методов, предназначенных для этого. Мы рассмотрим поочередно каждый из этих методов и разберемся в новых возможностях, которые можно использовать с объектами, которые не являются настоящими массивами, – например, с коллекциями элементов DOM и обычными объектами JavaScript.

В главе 11 рассматривается поддержка в Prototype основных элементов Web-приложений, а именно Document Object Model и HTML-форм. Функциональность, которую

мы рассмотрим в этой главе, значительно упрощает создание пользовательских интерфейсов, работающих в различных браузерах, и хорошо дополняет возможности, предоставляемые библиотекой Scriptaculous.

Часть IV завершает наше исследование этих библиотек парой дополнительных тем. В главе 12 мы вернемся к приложению QuickGallery, с которым работали в части I. Мы применим знания, полученные в частях II и III, чтобы быстро добавить в это приложение новые возможности. Эта глава иллюстрирует применение данных библиотек в реальном приложении.

Глава 13 посвящена интеграции Scriptaculous и Prototype с платформой Ruby on Rails. В ней показано, как Rails использует эти библиотеки для упрощения работы.

В основе всех Web-приложений лежит протокол HTTP, и если мы используем Ajax, то можем в определенной степени управлять работой браузера с этим протоколом. Однако возможность управлять накладывает на нас и определенную ответственность. В приложении А рассматриваются основы этого протокола, а в приложении Б — методики, которые мы используем в этой книге для измерения HTTP-трафика, генерируемого приложениями.

Основные примеры из этой книги полагаются на серверные части приложений, написанные на PHP и Java. Не все наши читатели знакомы с этими технологиями, и в приложениях В и Г мы шаг за шагом рассмотрим процесс установки соответствующих платформ — Java Tomcat и PHP/Apache — на основные операционные системы.

В этой книге мы делаем акцент в основном на JavaScript, и серверный код, который мы рассматриваем в некоторых главах, довольно прост. Однако мы используем несколько фокусов, которые, вероятно, будут необходимы в большинстве Ajax-приложений. В приложении Д мы покажем, как выполнять эти фокусы на нескольких популярных серверных языках.

Оформление фрагментов кода

Примеры кода, встречающиеся в этой книге, написаны на нескольких языках программирования, включая JavaScript, Java, JSP, Ruby и PHP. Объемистые примеры выделены в листинги, снабженные заголовками. Более короткие фрагменты просто вставлены в текст книги. Во всех случаях для оформления кода используется моноширинный шрифт с фиксированной шириной символов. Во многих длинных листингах есть пронумерованные метки, на которые даются ссылки в поясняющем тексте. Кроме того, иногда, если мы сравниваем несколько листингов, то изменения и отличия выделены **полужирным начертанием**.

Главы 8–11 посвящены низкоуровневым механизмам языка и библиотеки Prototype, и в них встречаются в основном небольшие фрагменты кода. Эти фрагменты можно выполнять в специальном приложении Scratchpad, входящем в состав скачиваемых примеров для этой книги. Это приложение позволяет следить за результатами выполнения данных фрагментов кода, и каждому фрагменту соответствует рисунок, на котором показаны результаты выполнения этого фрагмента.

Загружаемые примеры

Весь исходный код примеров для этой книги можно загрузить с сайта издательства Manning (<http://www.manning.com/crane3>). Среди этих примеров есть и интерактивный интерпретатор для выполнения фрагментов кода, и все примеры для каждой из

глав. Для работы некоторых из этих примеров необходимы Web-сервер с поддержкой Java или Apache/PHP. Как их установить, рассказывается в приложениях В и Г.

Что дальше?

Из нашей книги вы получите базовые знания о Prototype и Scriptaculous, но мы не можем ответить в ней на все вопросы, которые у вас возникнут. Издательство Manning предоставляет форум, позволяющий общаться с авторами книг. Вот его адрес: <http://www.manning-sandbox.com/forum.jspa?forumID=276>.

Кроме того, библиотекам Prototype и Scriptaculous посвящено изрядное количество онлайн-ресурсов. Например, официальный сайт документации Prototype находится по адресу <http://www.prototypejs.org>. Документация к библиотеке Scriptaculous доступна по адресу <http://wiki.script.aculo.us>. К тому же вопросы по обеим библиотекам можно задавать в группе Ruby on Rails Spinoffs, находящейся по адресу <http://groups.google.com/group/rubyonrails-spinoffs?hl=en>. Несмотря на название этой группы, в нее входят не только пользователи, работающие на Ruby, ей интенсивно пользуются и те, кто использует Prototype и Scriptaculous.

Общение с авторами

Покупка этой книги предоставляет вам доступ к закрытому форуму издательства Manning Publications, на котором вы можете комментировать эту книгу, задавать технические вопросы и получать помощь от авторов книги и других пользователей. Чтобы иметь доступ к форуму, обратитесь по адресу <http://www.manning.com/crane3>. На этой странице указано, как зарегистрироваться на форуме, как войти на него после регистрации, и изложены правила поведения на форуме.

Издательство Manning стремится обеспечить эффективный и удобный диалог между авторами книг и читателями. При этом общение авторов с читателями — сугубо добровольное (и не оплачиваемое).

Форум и архивы предыдущих обсуждений будут доступны, пока данная книга будет в продаже.

О названии

Сочетая описания, обзоры и примеры с последовательными описаниями действий, книги серии *In Action* помогают учиться и запоминать. Согласно результатам исследований, люди гораздо лучше запоминают информацию, полученную во время исследований, в которых они заинтересованы.

Хотя работники издательства Manning и не являются профессиональными психологами, мы убеждены, что для успешного обучения нужно пройти стадии исследования, игры и, самое важное, повторения уже изученного. Люди понимают и запоминают все новое, только если активно им пользуются. Люди учатся, делая. Характерная черта книг серии *In Action* — то, что они основаны на примерах. Это поощряет пользователя экспериментировать, применять разные варианты и писать новый код.

Есть и еще одна, более приземленная причина выбора такого названия для книги: наши читатели — всегда занятые люди. Книги нужны им, чтобы работать и решать практические задачи. Поэтому им нужны книги, которые позволяют сразу приступить к делу и учиться, уже делая что-то. Книги этой серии предназначены именно для таких читателей.

Об обложке

На обложке книги изображен житель Турции, облаченный в национальный костюм. Этот рисунок взят из книги о костюмах Османской империи, опубликованной 1 января 1802 года Уильямом Миллером из Лондона. У книги, из которой взят этот рисунок, не хватает обложки, и мы не смогли ее найти. В содержании книги приведена информация на английском и французском языках, и под каждой иллюстрацией указаны имена художников, которые работали над ней. Вероятно, эти люди сильно удивились бы, узнав, что их работа украшает обложку книги по программированию два столетия спустя.

Эту книгу о костюмах редактор издательства Manning купил на книжном развале на Уэст-стрит, 26, в Манхэттене. Продавец был американцем турецкого происхождения, и покупка была совершена, когда он уже складывал вещи, собираясь идти домой. У редактора не было с собой достаточно наличных для покупки, а чеки и кредитные карты явно не подходили для такого случая. Поскольку продавец собирался улететь в Анкару на следующее утро, ситуация становилась безнадёжной. Решение было найдено в форме устной договоренности, скрепленной рукопожатием. Продавец просто предложил переслать ему деньги электронным переводом, и редактор отправился домой с книгой под мышкой и банковскими реквизитами, записанными на клочке бумаги. Разумеется, мы перевели деньги на следующее утро, и до сих пор рады вере незнакомого нам человека в нашу надёжность.

Рисунок из книги об Османской империи демонстрирует богатство и разнообразие костюмов, существовавших двести лет назад. Он напоминает об изолированности и расстояниях, свойственных той эпохе, разительно отличающейся от нашего века молниеносных перемещений и универсализации.

За прошедшие два века изменились и стили одежды, и сами страны, и теперь часто бывает трудно отличить по одежде жителей разных континентов. Возможно, — если смотреть на это оптимистично, — мы променяли культурное разнообразие на более разнообразную личную жизнь. Или более разнообразную и интересную интеллектуальную и техническую жизнь.

Издательство Manning приветствует разнообразие, инициативу и новизну, свойственные компьютерным технологиям, и отдает дань этому разнообразию, поместив на обложку книги изображение костюма, созданное два века назад.