

Содержание

| | |
|--|-----------|
| Предисловие | 14 |
| Введение | 15 |
| Благодарности | 16 |
| Дэйв Крейн | 16 |
| Бэр Бибо | 16 |
| Об этой книге | 17 |
| Читатели | 17 |
| Структура книги | 17 |
| Оформление фрагментов кода | 19 |
| Загружаемые примеры | 19 |
| Что дальше? | 20 |
| Общение с авторами | 20 |
| О названии | 20 |
| Об обложке | 21 |
| От издательства | 22 |
| Часть I. Начало пути | 23 |
| Глава 1. Знакомство с Prototype и Scriptaculous | 25 |
| 1.1. Краткая история Ajax | 26 |
| 1.1.1. Доисторическая эпоха | 26 |
| 1.1.2. Эпоха первопроходцев | 27 |
| 1.1.3. Прибытие поселенцев | 27 |
| 1.1.4. Цивилизация | 28 |
| 1.2. Что такое Prototype | 28 |
| 1.2.1. Возможности языка JavaScript | 28 |
| 1.2.2. Возможности Web-браузеров | 30 |
| 1.3. Что такое Scriptaculous | 31 |
| 1.3.1. Визуальные эффекты | 32 |
| 1.3.2. Поддержка перетаскивания (drag and drop) | 32 |
| 1.3.3. Компоненты | 32 |
| 1.3.4. Утилиты и тестовые платформы | 33 |
| 1.4. Применение Prototype и Scriptaculous | 33 |
| 1.4.1. Ratings – приложение-пример | 33 |
| 1.4.2. Добавление Prototype и Scriptaculous | 39 |
| 1.5. Резюме | 44 |
| Глава 2. Web-приложение QuickGallery | 45 |
| 2.1. Дизайн и реализация | 46 |
| 2.1.1. Требования к приложению | 46 |
| 2.1.2. Перемещение по файловой системе | 47 |
| 2.1.3. Генерация уменьшенных копий изображений | 50 |
| 2.1.4. Объединение элементов приложения | 51 |
| 2.1.5. Создание HTML-страниц | 55 |

| | |
|--|------------|
| 2.2. Оценка классической модели Web-приложений | 58 |
| 2.2.1. Ссылки, формы и обновление страниц | 58 |
| 2.2.2. Классическая модель Web-приложений и QuickGallery | 59 |
| 2.3. Резюме | 60 |
| Глава 3. Упрощение Ajax с помощью Prototype | 61 |
| 3.1. Применение Ajax в приложении | 62 |
| 3.1.1. Проектирование асинхронных запросов | 62 |
| 3.1.2. Классы Ajax библиотеки Prototype | 65 |
| 3.2. Использование XML и Ajax | 70 |
| 3.2.1. Определение формата XML-ответа | 70 |
| 3.2.2. Модификация серверного кода | 71 |
| 3.2.3. Написание клиентского кода | 72 |
| 3.3. Использование JSON и Ajax | 80 |
| 3.3.1. Изменение серверного кода | 80 |
| 3.3.2. Изменение клиентского кода | 81 |
| 3.4. Резюме | 82 |
| Глава 4. Использование дополнительных возможностей Prototype по работе с Ajax | 83 |
| 4.1. Дополнительные Ajax-классы Prototype | 84 |
| 4.1.1. Класс Ajax.Updater | 84 |
| 4.1.2. Класс Ajax.PeriodicalUpdater | 85 |
| 4.1.3. Класс Ajax.Responders | 85 |
| 4.2. Использование HTML и Ajax | 86 |
| 4.2.1. Генерация HTML-фрагмента | 86 |
| 4.2.2. Изменение клиентского кода | 87 |
| 4.2.3. Обновление нескольких элементов DOM | 88 |
| 4.2.4. Автоматическое обновление содержимого | 93 |
| 4.3. Сравнение Ajax, основанного на данных, и Ajax, основанного на содержимом | 95 |
| 4.3.1. Простота разработки | 95 |
| 4.3.2. Инструменты разработки | 97 |
| 4.3.3. Сравнение производительности | 97 |
| 4.3.4. Возможности дальнейшего развития | 101 |
| 4.4. Резюме | 102 |
| Часть II. Библиотека Scriptaculous | 103 |
| Глава 5. Библиотека Scriptaculous Effects | 105 |
| 5.1. Быстрая победа: добавление эффекта одной строкой кода | 106 |
| 5.2. Настройка Scriptaculous и примеров | 107 |
| 5.2.1. Получение копии Scriptaculous | 108 |
| 5.2.2. Включение Scriptaculous на Web-страницы | 109 |
| 5.3. Типы эффектов Scriptaculous | 109 |
| 5.4. Эффекты – подробно | 110 |
| 5.4.1. Лаборатория эффектов | 111 |
| 5.4.2. Стандартные опции эффектов | 113 |
| 5.4.3. Базовые эффекты | 114 |
| 5.4.4. Комбинационные эффекты | 118 |

| | |
|--|------------|
| 5.5. Простое объединение эффектов сокрытия и отображения | 121 |
| 5.5.1. Вспомогательная функция <code>Effect.toggle()</code> | 122 |
| 5.5.2. Виджет <code>Toggle Pane</code> | 122 |
| 5.6. Работа с переходами | 128 |
| 5.6.1. Использование встроенных функций перехода | 128 |
| 5.6.2. Введение в <code>Transition Mapper</code> | 129 |
| 5.6.3. Создание функций перехода | 130 |
| 5.7. Получение управления во время выполнения эффектов | 133 |
| 5.8. Отмена эффектов | 134 |
| 5.9. Управление несколькими эффектами | 135 |
| 5.9.1. Параллельный запуск эффектов | 135 |
| 5.9.2. Поочередный запуск эффектов | 136 |
| 5.10. Резюме | 139 |
| Глава 6. Библиотека <code>Scriptaculous Controls</code> | 141 |
| 6.1. Использование программ-примеров из этой главы | 142 |
| 6.2. Элемент <code>InPlaceEditor</code> | 143 |
| 6.2.1. Создание элемента управления для редактирования текста на месте | 143 |
| 6.2.2. Параметры элемента управления <code>InPlaceEditor</code> | 146 |
| 6.2.3. Примеры применения | 153 |
| 6.3. Элемент управления <code>InPlaceCollectionEditor</code> | 160 |
| 6.3.1. Создание элемента <code>InPlaceCollectionEditor</code> | 160 |
| 6.3.2. Параметры элемента управления <code>InPlaceCollectionEditor</code> | 162 |
| 6.4. Элемент автозаполнения <code>Ajax</code> | 164 |
| 6.4.1. Создание элемента <code>Ajax.AutoCompleteer</code> | 165 |
| 6.4.2. Параметры элемента управления <code>Ajax.AutoCompleteer</code> | 167 |
| 6.4.3. Сервлет для элемента <code>Ajax.AutoCompleteer</code> | 170 |
| 6.4.4. Оформление вариантов автозаполнения | 174 |
| 6.5. Элемент <code>Autocomplete.Local</code> | 175 |
| 6.5.1. Создание элементов <code>Autocomplete.Local</code> | 176 |
| 6.5.2. Параметры элемента <code>Autocomplete.Local</code> | 178 |
| 6.6. Элемент управления <code>Control.Slider</code> | 180 |
| 6.6.1. Создание слайдера | 181 |
| 6.6.2. Использование графических файлов в оформлении слайдера | 184 |
| 6.6.3. Параметры элемента <code>Control.Slider</code> | 186 |
| 6.6.4. Более серьезный пример | 189 |
| 6.7. Резюме | 193 |
| Глава 7. Поддержка перетаскивания в библиотеке <code>Scriptaculous</code> | 195 |
| 7.1. Примеры кода для этой главы | 197 |
| 7.2. Перетаскивание объектов | 197 |
| 7.2.1. Преобразование элемента в перетаскиваемый | 198 |
| 7.2.2. Параметры класса <code>Draggable</code> | 200 |
| 7.3. Сбрасывание перетаскиваемых элементов | 206 |
| 7.3.1. Определение целей для сбрасывания | 206 |
| 7.3.2. Настройка целей для сбрасывания | 207 |
| 7.3.3. Пример перетаскивания | 211 |
| 7.4. Сортировка перетаскиванием | 216 |

| | |
|--|------------|
| 7.4.1. Создание сортируемого элемента | 216 |
| 7.4.2. Параметры сортируемых элементов | 218 |
| 7.4.3. Сериализация сортируемых элементов | 229 |
| 7.5. Резюме | 233 |
| Часть III. Библиотека Prototype | 235 |
| Глава 8. Все об объектах | 237 |
| 8.1. Приложение Scratchpad | 238 |
| 8.2. Тип Object | 239 |
| 8.2.1. Создание объектов | 240 |
| 8.2.2. Определение типов объектов с помощью прототипов | 242 |
| 8.2.3. Отражение объектов | 245 |
| 8.3. Объекты и библиотека Prototype | 247 |
| 8.3.1. Упрощение конструкторов с помощью метода Class.create() | 247 |
| 8.3.2. Объявление иерархий объектов с помощью метода Object.extend() | 249 |
| 8.3.3. Простое слияние объектов | 249 |
| 8.3.4. Моделирование иерархий объектов | 252 |
| 8.4. Резюме | 259 |
| Глава 9. Функции | 261 |
| 9.1. Функции в языке JavaScript | 262 |
| 9.1.1. Объявление функций | 262 |
| 9.1.2. Вызов функций | 264 |
| 9.1.3. Контекст функций | 265 |
| 9.1.4. Закрытые фрагменты кода | 266 |
| 9.1.5. Применение контекстов и закрытых фрагментов | 268 |
| 9.2. Расширение функций в Prototype.js | 270 |
| 9.2.1. Метод bind() | 270 |
| 9.2.2. Метод bindAsEventListener() | 271 |
| 9.3. Резюме | 272 |
| Глава 10. Массивы | 275 |
| 10.1. Знакомство с массивами | 276 |
| 10.2. Встроенный тип Array в JavaScript | 276 |
| 10.2.1. Перебор элементов массивов и свойство length | 276 |
| 10.2.2. Массивы как стеки — методы pop() , push() , shift() и unshift() | 277 |
| 10.2.3. Работа с блоками элементов — методы slice() , splice() и concat() | 278 |
| 10.2.4. Переупорядочивание массивов — методы reverse() и sort() | 280 |
| 10.2.5. Массивы и строки — методы join() и split() | 281 |
| 10.2.6. Массивоподобные объекты | 282 |
| 10.3. Массивы и библиотека Prototype | 282 |
| 10.3.1. Доступ к элементам по индексу с помощью методов first() , last() и indexOf() | 282 |
| 10.3.2. Изменение массивов с помощью методов clear() , compact() , without() и flatten() | 283 |
| 10.4. Методы объекта Enumerable | 284 |

| | |
|---|------------|
| 10.4.1. Поиск в массивах: <code>all()</code> , <code>any()</code> и <code>include()</code> | 286 |
| 10.4.2. Фильтрация массивов с помощью методов <code>detect()</code> , <code>findAll()</code> , <code>reject()</code> , <code>grep()</code> и <code>partition()</code> | 287 |
| 10.4.3. Обработка сложных элементов массивов — методы <code>pluck()</code> , <code>invoke()</code> и <code>collect()</code> | 290 |
| 10.4.4. Создание сложных элементов массивов с помощью методов <code>inject()</code> и <code>zip()</code> | 294 |
| 10.4.5. Упорядочение числовых массивов — методы <code>max()</code> , <code>min()</code> и <code>sortBy()</code> | 297 |
| 10.4.6. Работа с коллекциями DOM | 298 |
| 10.5. Работа с объектами <code>Hash</code> и <code>ObjectRange</code> | 299 |
| 10.5.1. Использование объектов <code>Hash</code> | 299 |
| 10.5.2. Использование типа <code>ObjectRange</code> | 303 |
| 10.6. Резюме | 303 |
| Глава 11. Возвращение к браузеру | 305 |
| 11.1. Краткий обзор методов DOM | 305 |
| 11.1.1. Перемещение по дереву элементов | 307 |
| 11.1.2. Изменение элементов дерева | 309 |
| 11.2. Prototype и DOM | 312 |
| 11.2.1. Простые вспомогательные функции | 312 |
| 11.2.2. Объект <code>Element</code> | 314 |
| 11.2.4. Объект <code>Insertion</code> | 320 |
| 11.2.4. Объекты <code>Position</code> | 321 |
| 11.3. Расширение объекта <code>Event</code> | 325 |
| 11.4. Работа с HTML-формами | 327 |
| 11.5. Резюме | 332 |
| Часть IV. Дополнительные темы | 333 |
| Глава 12. Prototype и Scriptaculous на практике | 335 |
| 12.1. Требования к приложению <code>QuickGallery</code> | 335 |
| 12.2. Создание редактора слайд-шоу | 336 |
| 12.2.1. Модификация структуры страницы | 337 |
| 12.2.2. Определение стратегии перетаскивания | 341 |
| 12.2.3. Реализация перетаскивания | 343 |
| 12.2.4. Реализация редактируемых подписей | 346 |
| 12.3. Реализация сохраняемости с помощью Ajax | 349 |
| 12.3.1. Определение формата хранения данных | 349 |
| 12.3.2. Сохранение слайд-шоу | 349 |
| 12.3.3. Загрузка данных с сервера | 351 |
| 12.4. Создание панели инструментов | 353 |
| 12.5. Создание механизма воспроизведения слайд-шоу | 356 |
| 12.5.1. Описание пользовательского интерфейса | 356 |
| 12.5.2. Реализация механизма воспроизведения слайд-шоу | 358 |
| 12.5.3. Запуск механизма воспроизведения слайд-шоу | 361 |
| 12.6. Обзор кода | 364 |
| 12.7. Резюме | 379 |

| | |
|--|-----|
| Глава 13. Prototype, Scriptaculous и Rails | 381 |
| 13.1. Генерация кода на JavaScript | 382 |
| 13.1.1. Программа “Hello, World” в стиле Prototype и Rails | 382 |
| 13.1.2. Вспомогательные методы Rails | 383 |
| 13.2. Вспомогательные методы для Ajax | 385 |
| 13.2.1. Задание стандартных настроек | 386 |
| 13.2.2. Обзор вспомогательных методов для работы с Ajax | 387 |
| 13.3. Вспомогательные методы для Scriptaculous | 393 |
| 13.3.1. Создание визуальных эффектов | 393 |
| 13.3.2. Реализация редактирования на месте | 394 |
| 13.3.3. Добавление возможностей автозаполнения | 396 |
| 13.3.4. Реализация перетаскивания | 397 |
| 13.4. Использовать JavaScript или нет | 402 |
| 13.5. Следующий уровень: RJS | 402 |
| 13.5.1. Проблема множественных обновлений | 403 |
| 13.5.2. “Hello, World” в стиле RJS | 404 |
| 13.5.3. Множественные обновления и RJS | 405 |
| 13.5.4. Обзор RJS | 406 |
| 13.5.5. Встраиваемый RJS | 409 |
| 13.6. Резюме | 409 |
| Приложение А. Введение в HTTP | 411 |
| А.1. Почему нас должен интересовать HTTP | 411 |
| А.2. Что такое HTTP | 412 |
| А.2.1. Ресурсы и URL | 412 |
| А.2.2. Отсутствие состояний | 413 |
| А.3. Что такое типы запросов | 413 |
| А.4. Форматы HTTP-сообщений | 414 |
| А.4.1. Формат сообщений запросов | 414 |
| А.4.2. Формат сообщений ответов | 416 |
| А.5. Что такое Ajax-соединения | 417 |
| А.5.1. Генерация запросов с помощью Ajax | 417 |
| А.5.2. Получение ответов с помощью Ajax | 418 |
| А.5.3. Страница-лаборатория для работы с запросами | 419 |
| Приложение Б. Измерение HTTP-трафика | 423 |
| Б.1. Сбор данных об HTTP-трафике | 423 |
| Б.1.1. Mozilla LiveHTTPHeaders | 424 |
| Б.1.2. Microsoft Fiddler | 425 |
| Б.2. Обработка данных по HTTP-трафику | 427 |
| Приложение В. Установка и запуск Tomcat 5.5 | 433 |
| В.1. Получение и распаковка дистрибутива | 433 |
| В.2. Переменная JAVA_HOME | 434 |
| В.3. Настройка контекстов приложений | 435 |
| В.4. Запуск Tomcat | 435 |
| В.5. Управление контекстами Tomcat | 436 |

| | |
|--|-----|
| Приложение Г. Установка и запуск PHP | 439 |
| Г.1. Требования к системе | 439 |
| Г.1.1. Версия PHP | 439 |
| Г.1.2. Web-сервер | 440 |
| Г.1.3. Операционная система | 440 |
| Г.1.4. Библиотеки | 440 |
| Г.2. Установка | 440 |
| Г.2.1. Установка PHP под Linux | 440 |
| Г.2.2. Установка PHP под Mac OS X | 442 |
| Г.2.3. Установка PHP под Windows | 444 |
| Г.3. Конфигурирование QuickGallery | 445 |
| Приложение Д. Портирование серверных механизмов | 449 |
| Д.1. Чтение заголовков запросов | 449 |
| Д.1.1. PHP | 450 |
| Д.1.2. Java Servlet API | 450 |
| Д.1.3. .NET | 450 |
| Д.1.4. Ruby on Rails | 451 |
| Д.2. Добавление заголовков в ответы | 451 |
| Д.2.1. PHP | 451 |
| Д.2.2. Java Servlet API | 451 |
| Д.2.3. Java Servlet Pages | 452 |
| Д.2.4. .NET | 452 |
| Д.2.5. Ruby on Rails | 452 |
| Д.3. Чтение тела POST-запроса | 453 |
| Д.3.1. PHP | 453 |
| Д.3.2. Java Servlet API | 453 |
| Д.3.3. .NET | 454 |
| Д.3.4. Ruby on Rails | 454 |
| Д.4. Резюме | 454 |
| Предметный указатель | 455 |