

Содержание (сводка)

	Введение	21
1	Первые шаги в разработке веб-приложений. <i>Адаптивный веб-дизайн</i>	33
2	Важная адаптивность. <i>Адаптивный веб-дизайн: сперва проектируем для мобильной платформы</i>	75
3	Отдельный сайт для мобильных устройств. <i>Сталкиваемся с непредвиденными обстоятельствами</i>	123
4	Решение о том, кого поддерживать. <i>Поддержку каких устройств нам следует обеспечить?</i>	169
5	Базы данных и классы устройств. <i>Распределение по группам</i>	183
6	Создание мобильного веб-приложения с помощью фреймворка. <i>Приложение Tartanator</i>	249
7	Мобильные веб-приложения в реальном мире. <i>Мобильные суперприложения</i>	299
8	Создание гибридных мобильных приложений с помощью PhoneGap. <i>Приложение Tartan Hunt: пишем нативный код</i>	345
9	Как быть ориентированным на будущее. <i>Разбираемся в хаосе</i>	389
I	Напоследок. <i>Шесть важных тем (которые мы не рассмотрели)</i>	405
II	Конфигурирование среды вашего веб-сервера. <i>Нужно же где-то начинать</i>	419
III	Установка WURFL. <i>Детектируем устройства</i>	429
IV	Установка Android SDK и инструментов. <i>Заботимся о среде</i>	435

Содержание (полное)

Введение

Вы хотите научиться разрабатывать веб-приложения. Когда вы пытаетесь чему-либо научиться, мозг постоянно следит за тем, чтобы процесс обучения не остановился. Ваш мозг говорит: «Лучше оставить место для более важных сведений, например, каких диких животных следует избегать». Так как же убедить свой мозг в том, что умение разрабатывать мобильные веб-приложения не менее важно для вас?

Для кого написана эта книга?	22
Мы знаем, о чем вы думаете	23
Мы также знаем, о чем думает ваш мозг	23
Метапознание: учимся учиться	25
Технические рецензенты	30
Благодарности	31
От издательства	32

1 первые Шаги В разработке Веб-приложений

Адаптивный веб-дизайн

Вы готовы погрузиться в разработку веб-приложений для мобильных устройств? Здесь присутствует особая магия и увлекательность, а также масса моментов, когда хочется воскликнуть *«Эврика!»*. Однако, помимо этого, имеются свои загадки и неразбериха. Мобильные технологии развиваются с ошеломляющей скоростью, и знать приходится так много! Держитесь крепче. Первым делом мы продемонстрируем вам подход к созданию веб-сайтов, называемый **адаптивным веб-дизайном** (Responsive Web Design, RWD). Вы сможете приспособить сайты таким образом, чтобы они отлично выглядели на многих мобильных устройствах, взяв за основу имеющиеся у вас навыки веб-разработки.

index.html



styles.css



Переходим на мобильную платформу	34
По дороге в паб произошло кое-что необычное	36
Если браузеры в мобильных телефонах такие замечательные...	37
Чем же так сильно различаются мобильные веб-приложения?	38
Адаптивный веб-дизайн	42
Разный CSS в разных местах	44
Медиазапросы CSS	45
Текущая структура сайта The Splendid Walrus	47
Анализ текущего CSS	48
Что необходимо изменить?	49
Идентифицируем CSS, который необходимо изменить	50
Этапы создания мобильно-специфичного CSS	51
Так что же плохого в макете с фиксированной шириной?	58
Насколько лучше макет резиновой сетки?	59
Формула резинового макета	60
Продолжаем преобразование в резиновый макет	61
Переключение контекста	63
Что не так с данным изображением?	64
Резиновые изображения	65
Не забывайте быть ответственным	68
Это адаптивный сайт!	72
Адаптивный веб-дизайн — это также настрой ума	73

2

Важная адаптивность

Адаптивный веб-дизайн: сперва проектируем для мобильной платформы

Красивый сайт для мобильных устройств. Однако красота — лишь оболочка. Он может выглядеть как сайт для мобильных устройств, но по-прежнему оставаться сайтом для настольных устройств в «мобильном обмундировании». Если мы хотим, чтобы сайт быстро работал на мобильных устройствах, нужно начать с проектирования для мобильной платформы. Мы начнем с «вскрытия» текущего сайта и поиска «настольного скелета» в его «мобильном шкафу». Наведем порядок и начнем все заново с использованием концепции **прогрессивного улучшения**, пройдя путь от базового содержимого до настольного представления. В итоге вы получите страницу, оптимизированную независимо от размера экрана.

Прогрессивное улучшение, основанное на размере экрана и клиентских функциях

Очень маленькие экраны
(мобильные телефоны)



Маленькие экраны
(смартфоны)



Экраны среднего размера
(планшетные компьютеры)



Большие экраны
(настольные мониторы и телевизоры)



Стоило вам только подумать о том, что пришло время праздновать...	76
А есть ли проблема на самом деле? Как нам это узнать?	77
Что делать, если все работает не очень быстро	79
Не дайте внешнему виду ввести вас в заблуждение — это ТЯЖЕЛАЯ страница	80
Ценные файлы HAR	81
Поиск помех для скорости загрузки страницы	83
Откуда берется JavaScript, относящийся к Google Maps?	85
Выглядит дружелюбным к мобильным устройствам, но это не так	87
Адаптивный веб-дизайн: сперва проектируем для мобильной платформы	88
Что такое прогрессивное улучшение?	89
Корректировка плавающего содержимого	92
Медиазапросы «сперва мобильная версия»	93
Сюрприз! Страница некорректно отображается в Internet Explorer	94
Один атрибут src для управления всеми изображениями	100
Увеличение в метатеге viewport	104
Право изменять масштаб?	105
Возвращение карты обратно с использованием JavaScript	106
Написание псевдомедиазапроса на JavaScript	108
Добавление JavaScript в страницу On Tap Now	109
Данные виджеты неадаптивны	111
Перенос и сопоставление атрибутов iframe с CSS-эквивалентами	112
Удаление атрибутов из JavaScript	113
Карта снова перекрывает изображения	115
Пусть содержимое станет вашим гидом	116
Точки останова приходят на помощь	119

3

отдельный сайт для мобильных устройств

Сталкиваемся с неподвижными обстоятельствами

Мечта о едином Интернете, где властвовала бы адаптивность, прекрасна... Мечта, в которой у всех сайтов имеется один макет, созданный по принципу «сперва проектируем для мобильной платформы». Но что делать, когда сталкиваешься с неприятной реальностью: например, с устаревшими системами или устройствами либо с ограничениями, налагаемыми бюджетом заказчика? В этой главе мы рассмотрим подробности **детектирования мобильных пользователей, поддержки устаревших телефонов и создания отдельного сайта для мобильных устройств.**



У Creature Comforts есть агенты, работающие в полевых условиях	124
Как агенты получают необходимую информацию и делятся ею?	125
Направляйте мобильных пользователей на сайт, оптимизированный под мобильные устройства	128
Анализ с целью выявления мобильных пользователей	129
Знакомство с пользовательскими агентами	130
Пользовательские агенты — порождение дьявола?	133
Разговор начистоту: большинство крупнейших сайтов имеет версию для мобильных устройств	136
Когда вам действительно нужно организовать (пере-)направление	137
Взгляните на сценарий	138
Как работает данный сценарий?	139
Создание мобильного оригинал-макета	140
Срочная доставка... осложняющих факторов	142
Не все телефоны являются смартфонами... далеко не все	145
Будем придерживаться простоты: встречайте XHTML-MP	146
Зачем нам может потребоваться эта старая штука?	147
Используйте XHTML-MP, чтоб комар носа не подточил	148
Кстати, прокрутка — это очень неудобно	151
Наш последний финт	151
Клавиши доступа в действии	155
Где были допущены ошибки?	156
Исправление ошибок	157
Поддерживающий мобильные устройства CSS	159
Хммм... чего-то не хватает	164
Очень не хватает вида кнопок!	165
Большой успех!	166

4

решение о том, кого поддерживать

Поддержку каких устройств нам следует обеспечить?

Целого дня не хватит для тестирования на каждом из устройств. Вам придется определиться, поддержку каких устройств вы сможете обеспечить. **Но как решить, где именно провести черту?** Что будет с пользователями устройств, тестирование на которых у вас нет возможности осуществить, — они останутся за бортом? В этой главе мы смешаем магическое зелье из **требований проекта** и **предпочтений целевой аудитории**, которое поможет понять, **поддержку каких устройств нам следует обеспечить** и **что делать с теми, которые не поддерживаются.**

Как узнать, где именно следует провести черту?	170
Отвлекитесь ненадолго от клавиатуры	171
Вещи, которые вы не поддерживаете, против вещей, поддержку которых не можете обеспечить	172
Ставьте вопросы по своему проекту	174
Ингредиенты нашего магического зелья	176
Инструментарий и данные, помогающие принять решение	177
Как узнать, что у моих клиентов имеются подходящие телефоны?	182

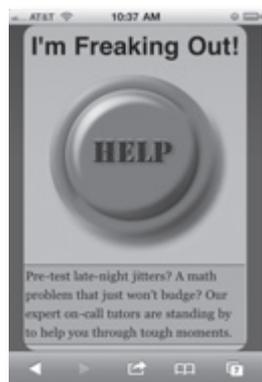


5

базы данных и классы устройств

Распределение по группам

При задании планки для устройств, которые мы станем поддерживать, не учитывается ряд проблем. Как узнать достаточно информации о мобильных браузерах пользователей, чтобы понять, отвечают ли они соответствующим требованиям, перед тем, как доставлять им содержимое? Как избежать генерирования («хромающего») содержимого ради наименьшего общего знаменателя? Как организовать все это, не сойдя с ума? В этой главе мы затронем область **функциональных возможностей устройств**, научимся получать к ним доступ с помощью **базы данных устройств** и, наконец, узнаем, как группировать их по **классам устройств**.



Тревожная кнопка для впавших в панику студентов	184
Источники данных о мобильных устройствах приходят на помощь	186
Знакомство с WURFL	187
База данных WURFL и ее возможности	188
WURFL: смысловой код API-интерфейса	191
Мы можем также создать исследовательскую страницу	192
Исследовательская страница: создание среды	193
Быстро вносим небольшие изменения для улучшения нашей исследовательской страницы	200
Задействуем информацию о функциональных возможностях	202
Использование WURFL для разграничения содержимого	202
Инициализация объекта device и подготовка данных	204
Является ли данное устройство мобильным?	204
Делаем страницу немного умнее с помощью WURFL	208
Кнопка тревоги: только для телефонов	209
Классы устройств	213
Взгляд на домашнюю страницу в адаптированном под мобильные устройства варианте	215
Группировка требований согласно разным типам мобильных устройств	216
Классы устройств, которые у нас будут в итоге	217
Знакомство с функцией сопоставления	223
Что происходит в операторе switch?	224
Проверка функциональных возможностей с помощью функции сопоставления	225
Заполнение пробелов в тестах, касающихся классов устройств	232
Нам необходима более внушительная страховка	243
Своевременное принятие мер	244

6 Создание мобильного Веб-приложения с помощью фреймворка

Приложение Tartanator

«Нам требуется приложение!» Всего пару лет назад этот клич означал только одно: разработку нативного кода и развертывание для каждой платформы, которую вам требовалось поддерживать. Однако нативный код — не единственный оптимальный вариант. Веб-приложения для мобильных браузеров заслужили доверие среди широких слоев населения — особенно сейчас, когда модный **HTML5** вместе с **CSS3** и **JavaScript** в ударе. Окунемся в мир мобильных веб-приложений, для чего возьмем **фреймворк для разработки мобильных веб-приложений** — программные инструменты, призванные помочь вам быстро выполнить работу, — и опробуем его!

Хм... неплохо.
А можно сделать так,
чтобы все это было
больше похоже
на... нативное
приложение?



Приложение... HTML5... что означают эти понятия?	251
Каково обычное поведение «традиционных» веб-сайтов	252
Каково обычное поведение сайтов, похожих на приложения	253
Общий план первой стадии работ над Tartanator	256
Зачем использовать фреймворки для создания мобильных веб-приложений?	257
Наш выбор для Tartanator: jQuery Mobile	258
Создание базовой страницы с помощью jQuery Mobile	260
Разметка остальной части страницы	261
HTML5-атрибут data*	263
Связывание с множественными страницами посредством jQuery Mobile	266
Улучшение внешнего вида списка	273
Добавление остальных клетчатых узоров	275
Фильтрация и упорядочение списка	276
Добавление панели инструментов в нижний колонтитул	281
Придание красивого внешнего вида панели инструментов	282
Завершение работы над структурой	283
Пора соорудить форму для создания клетчатых узоров на ткани	285
Форма для создания клетчатых узоров	287
Создание формы HTML5	288
Пришло время добавить базовые поля	289
Списки внутри списков позволят пользователям добавлять цвета	290
Компонентные пары «цвет — размер»: поле для выбора цвета	291
Пары полей для ввода значений цвета и размера: поле для указания размера	292
Ссылка, ведущая к форме	294

7

Мобильные Веб-приложения В реальном Мире

Мобильные суперприложения

Мобильное веб-приложение напоминает одаренного ребенка. Такого, который способен на удивительные вещи, но в то же время является непредсказуемым смутьяном. Мы старались сдерживать сидящего в нем гиперактивного гения, вводя ограничения и устанавливая рамки, но пришло время извлечь выгоду из некоторых природных талантов мобильного веб-приложения. Мы можем прибегнуть к **прогрессивному улучшению**, чтобы принарядить интерфейс в продвинутых браузерах и трансформировать невозможность постоянного подключения из бремени в простую особенность путем обеспечения поддержки тактичного **автономного режима**. И мы сможем постичь суть мобильности путем использования **API-интерфейса Geolocation**. Давайте превратим это веб-приложение в мобильное суперприложение!



Выглядит неплохо...	300
Мобильные приложения в реальном мире	302
На старт, внимание, улучшаем!	306
Улучшение формы	307
Виджет для управления списком цветов и размеров	308
Заглядываем внутрь	309
Итак, это были улучшения на стороне клиента...	310
...а теперь займемся стороной сервера	312
Две стороны generate.php	313
Осталось сделать еще одну вещь!	314
Возможность работать в автономном режиме имеет важное значение	316
Базовый рецепт создания манифеста кэша	317
Инструменты разработки как спасительное средство	318
Загрузка файла манифеста кэша с использованием корректного Content-type	319
Мы (наконец-то) одержали победу	329
Как работает геолокация	330
Как получить у W3C-совместимых браузеров сведения об их местоположении	331
Начинаем работу над страницей Find Events (Поиск событий): обеспечение базового взаимодействия	333
Интегрируем геолокацию	335
Ничего не найдено	341

8

создание Гибридных Мобильных приложений с помощью PhoneGap

Приложение Tartan Hunt: пишем нативный код

Иногда приходится писать нативный код. Например, если вам требуется доступ к чему-то недоступному (пока что) в мобильных браузерах либо вашему клиенту *нужно* разместить приложение в App Store. Мы заглядываем в светлое будущее, когда браузер будет иметь доступ ко всему, что мы пожелаем, а мобильные веб-приложения будут обладать привлекательной функциональностью, которой располагают нативные приложения. А до тех пор у нас есть возможность выбора **гибридной разработки** — мы продолжим писать код с применением веб-стандартов и воспользуемся **библиотекой для ликвидации брешей** между нашим кодом и нативными возможностями устройств. **Кросс-платформенные нативные приложения, созданные с применением веб-технологий,** — не такой уж плохой компромисс!



Нам снова предоставилась возможность поработать	346
Как работают гибридные приложения?	349
Устранение брешей между веб-приложением и нативным кодом с помощью PhoneGap	350
Знакомство с PhoneGap Build	353
Как будет работать приложение Tartan Hunt?	354
Учет найденных клетчатых узоров	355
Анатомия проекта Tartan Hunt	356
Загрузка приложений на компьютер	360
Выбор устройства для установки приложения	361
Кто что видел? Сохранение изображений найденных узоров	366
Что localStorage может для нас сделать?	367
Проверка того, что поддерживает браузер	371
Использование функции для показа того, какие узоры уже найдены	372
Методы toggle и toggleClass	373
Вы нашли клетчатый узор? Докажите это!	376
Вовлечение PhoneGap в процесс фотосъемки	377
PhoneGap почти готов предстать крупным планом	379
Теперь мы готовы перейти к API-интерфейсу mediaCapture	380
Как будет осуществляться обработка, если все пройдет успешно?	381
В реальной жизни все выглядит немного иначе	382
Анонимная функция	383
И еще кое-что напоследок!	385
Мы справились!	386





КАК БЫТЬ ОРИЕНТИРОВАННЫМ НА БУДУЩЕЕ

Разбираемся в хаосе

Адаптивный веб-дизайн. Детектирование устройств. Мобильные веб-приложения. PhoneGap. Чему нам отдать предпочтение?

Существует множество подходов к разработке веб-приложений для мобильных устройств. Зачастую в проектах задействуется **несколько подходов**. Единственного правильного решения не существует. Но не стоит беспокоиться. Самое важное — научиться плыть по течению. **Учитывайте неопределенность**. Перенимайте **ориентированный на будущее настрой** и скользите по волнам, пребывая в уверенности, что вы способны проявить гибкость и готовы адаптироваться к тому, что может преподнести будущее.



И что теперь?	390
Пора развеять заблуждения насчет контроля	393
Манифест Future Friendly	394
Не можете быть полностью готовыми к будущему, будьте хотя бы ориентированными на будущее	396
Сегодня приложение, завтра — веб-страница	397
Это длинный путь: вот некоторые ориентиры	398
Смешивание мобильных «ингредиентов»	401
Ориентируйтесь на будущее	403

ЛАЗЕРНЫЙ ФОКУС

У нас нет возможности позаботиться обо всем на **всех** устройствах. Чтобы существовать в мире постоянно возрастающей сложности устройств, необходимо сфокусироваться на том, что имеет наибольшее значение для клиентов и нашего дела. Мы должны создавать не решения, которые удовлетворяют неэффективных пользователей, а осмысленное содержимое и эффективные службы. Кроме того, люди все больше устали от возни и поисков путей упрощения для себя определенных вещей. Сфокусируйтесь на том, что имеет существенное значение, прежде чем это за вас сделают клиенты и увеличивающееся разнообразие сторонних решений.

ОРБИТА ВОКРУГ ДАННЫХ

Экосистеме устройств необходимо быть интероперабельной, а наиболее легким способом обеспечить это является устойчивый обмен данными. Реагируйте на уже имеющиеся и появляющиеся возможности определением своих данных путем, который:

- делает возможными множественные (гибкие) формы доступа и уведомлений;
- задействует стандарты для обеспечения интероперабельности;
- сфокусирован на долговременной целостности;
- включает важные и перманентные ссылки на все содержимое;
- поддерживает операции чтения/записи.

УНИВЕРСАЛЬНОЕ СОДЕРЖИМОЕ

Хорошо структурированное содержимое является неотъемлемой частью процесса разработки. Поразмыслите над тем, как оно будет вписываться в разнообразные контейнеры, не забывая при этом об их ограничениях и возможностях. Не бойтесь исследовать новые направления.

Мощные смарт-устройства, простые и ограниченные в функционале устройства, интероперабельные устройства и (многие) другие аппараты являются частью нашего будущего. Структурируйте и сохраняйте свое содержимое соответствующим образом.

НЕИЗВЕСТНЫЙ КОРАБЛЬ, ТРЕБУЕТСЯ ИДЕНТИФИКАЦИЯ

Реагирование на каждую вариацию устройств делает инклюзивный дизайн очень сложным. Довольно неплохой высокоуровневый набор стандартов для типов устройств способен упростить процесс адаптации. Подробная профильная информация может выступать дополнением к данным стандартам.

Систематизация классов устройств обеспечивает для производителей определенное ориентиры, одновременно учитывая, что завтра возможно появление новых типов устройств.

КОМАНДУЙТЕ СВОИМ ФЛОТОМ

Располагая широким спектром устройств, мы получаем возможность распределять между ними задачи и информацию. При управлении взаимодействиями в рамках набора устройств на каждое из них можно будет возложить обеспечение именно того взаимодействия, на которое оно окажется лучше всего способно. Это устраняет необходимость приспособлять все аспекты той или иной службы под каждое устройство и позволяет нам оперировать в рамках экосистемы функциональных возможностей устройств.

I напоследок

Шесть важных тем (которые мы не рассмотрели)

У вас когда-нибудь было ощущение, что чего-то не хватает? Мы вас прекрасно понимаем... Казалось бы, книга закончилась, однако есть еще кое-что. Мы не могли оставить вас без нескольких дополнительных деталей, которые просто не уместились в книгу (иначе вам пришлось бы перевозить эту книгу на тележке, настолько она была бы тяжелой). Итак, давайте посмотрим, чего (все еще) не хватает в данном издании.

1. Тестирование на мобильных устройствах	406
2. Удаленная отладка	408
3. Какие браузеры что поддерживают	414
4. API-интерфейсы устройств	416
5. Магазины приложений и распространение программных продуктов	417
6. Адаптивный веб-дизайн и серверные компоненты – RESS (REsponsive design + Server-Side components)	418

II конфигурирование среды Вашего Веб-сервера

Нужно же где-то начинать

Писать или произносить фразу «веб-приложения для мобильных устройств» без приставки «веб» в нашем случае не совсем правильно. Если вы хотите разрабатывать веб-приложения для мобильных устройств, вам понадобится веб-сервер. Иного пути нет. Вам нужно где-то разместить свои онлайн-данные, воспользовавшись платными услугами веб-хостинга, дата-центром корпоративного уровня либо собственным компьютером. В данном приложении мы рассмотрим процедуру **установки локального веб-сервера** на вашем компьютере, а также **позаботимся о PHP-среде**, используя бесплатное ПО с открытым исходным кодом.

Что нам нужно от вас	420
Доступность лишь локально	421
Windows и Linux: установка и конфигурирование XAMPP	422
Начало работы с XAMPP	423
Для пользователей Mac OS X: пришло время MAMP	424
Убедитесь в правильности выбранного порта	425
Доступ к вашему веб-серверу	426
phpInfo, пожалуйста!	428

III

Установка WURFL

Детектируем устройства

На первом этапе разгадки тайны детектирования устройств придется **немного побегать**. Любой хороший сыщик знает, что необходимо собрать улики и опросить свидетелей. Сначала нам нужно найти «мозг» операции — **API-интерфейс PHP WURFL**. Затем мы займемся «мускулами» — информацией о функциональных возможностях тысяч устройств, собранной в одном **файле данных XML**. Однако нам придется немного надавить, чтобы они рассказали нам всю историю, поэтому мы осуществим небольшое **конфигурирование**, а также уделим пристальное внимание некоторым аспектам.

Где можно найти «мозг» WURFL?	430
А где найти «мускулы»?	431
Обеспечение совместной работы	432
Небольшие служебные операции с файловой системой	433
Обратите внимание!	434

IV

Установка Android SDK и инструментов

Заботимся о среде

Чтобы стать мастером в тестировании нативных **Android-приложений**, вам необходимо **ориентироваться в соответствующей среде**. Вам потребуется превратить свой компьютер в небольшую экосистему, в которой вы сможете выводить **Android-приложения** «на выпас» на виртуальные (эмулированные) либо на настоящие устройства. Чтобы сделать из вас «пастуха» **Android-«овец»**, мы расскажем, где можно скачать **набор средств для разработки программного обеспечения Android SDK**, как установить пакет **Platform Tools**, как **создавать виртуальные устройства** и как **устанавливать и удалять приложения**.

Давайте скачаем Android SDK	436
Обзаводимся подходящими для работы инструментами	437
Создание нового виртуального устройства	440
Поиск правильного пути	445