

Using HTML5 to Design Web Apps for Tablets and Smartphones



Free Sampler

jQuery Mobile

Up and Running



Maximiliano Firtman

Want to read more?

You can [buy this book](#) at [oreilly.com](#)
in print and ebook format.

Buy 2 books, get the 3rd FREE!

Use discount code: OPC10

All orders over \$29.95 qualify for **free shipping** within the US.

It's also available at your favorite book retailer,
including the iBookstore, the [Android Marketplace](#),
and [Amazon.com](#).



O'REILLY®

Spreading the knowledge of innovators

[oreilly.com](#)

jQuery Mobile: Up and Running

by Maximiliano Firtman

Copyright © 2012 Maximiliano Firtman. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editor: Simon St. Laurent

Indexer: Lucie Haskins

Production Editors: Jasmine Perez and Teresa Elsey

Cover Designer: Karen Montgomery

Copyeditor: Linley Dolby

Interior Designer: David Futato

Proofreader: Stacie Arellano

Illustrator: Robert Romano

February 2012: First Edition.

Revision History for the First Edition:

2012-02-07 First release

See <http://oreilly.com/catalog/errata.csp?isbn=9781449397654> for release details.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *jQuery Mobile: Up and Running*, the image of a Japanese sable, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-449-39765-4

[LSI]

1328624859

Table of Contents

Preface	ix
1. The Mobile Platform	1
Why Do We Need jQuery Mobile?	1
Myths of the Mobile Web	1
Mobile Webapps	3
So, Again...Why Do We Need jQuery Mobile?	3
What Is jQuery Mobile?	4
What jQuery Mobile Is Not	5
The Framework	5
The Mobile and Tablet World	6
Device Categories	6
Operating Systems and Browsers	9
jQuery Mobile Compatibility	11
HTML5 and CSS3	14
Main Features	15
Use of Nonintrusive Semantic HTML5	16
Progressive Enhancement	17
Accessibility Support	18
Testing Webapps	18
Emulators and Simulators	19
Remote Labs	23
2. Starting with the Framework	25
Preparing the Document	25
Requirements	25
Hosting the Files	25
Using a CDN	27
Main HTML5 Template	29
Adobe Dreamweaver Support	32
Previewing Files	33

Architecture	34
Roles	34
Theming	35
The Page	36
Navigation	40
Back Button	41
Internal Page Links	42
External Page Links	44
Absolute External Links	49
Mobile Special Links	50
Transition Between Pages	50
Reverse Transition	52
Dialogs	52
Closing or Going Back?	54
Opening Pages from Dialogs	56
Integrating with the Phone	57
Making a Call	58
Video and VoIP Calls	59
Sending Email	60
Sending an SMS	60
Other URI Schemes	61
Bringing It All Together	61
3. UI Components	63
Toolbars	63
Positioning	64
True Fixed Toolbars	66
Adding Content to the Header	66
Adding Content to the Footer	69
Navigation Bars	70
Persistent Footer	73
Formatting Content	75
Collapsible Content	76
Accordion	79
Columns	81
Buttons	82
Inline Buttons	83
Grouped Buttons	83
Effects	85
Icons	85
Creating Custom Icons	87
Icon Positioning	88
Icon-Only Buttons	89

Icon Shadow	89
4. Lists	91
Full-Page Lists Versus Inset Lists	94
Visual Separators	96
Interactive Rows	98
Nested Lists	101
Split Button Lists	104
Ordered Interactive Lists	107
Using Images	107
Row Icons	107
Thumbnails	108
Aside Content	109
Title and Description	111
Using Count Bubbles	111
Filtering Data with Search	112
List Views Cheat Sheet	114
5. Form Components	115
Form Action	115
Forcing a Non-AJAX Form	116
Form Elements	116
Labels	117
Field Containers	117
Text Fields	118
Auto-Growing Text Area	119
New HTML5 Attributes	121
Date Fields	122
Slider	123
Flip Toggle Switch	124
Select Menus	125
Radio Buttons	135
Checkboxes	137
File Uploads	139
6. The Framework and JavaScript	141
Document Events	141
Configuration	143
Global Configuration	144
Page Configuration	149
Widgets Configuration	150
Utilities	152
Data-* Utilities	152

Page Utilities	152
Platform Utilities	155
Path Utilities	155
UI Utilities	155
Custom Transitions	156
Dynamic Content	157
Creating Pages	157
Creating Widgets	160
Updating Widgets	160
Creating Grids	162
Changing Page Contents	162
Event Handling	162
Page Events	162
Widget Events	165
Orientation Event	165
Gesture Events	166
Virtual Clicks Events	167
7. Creating Themes	169
ThemeRoller	170
Global Settings	171
Swatch Color Settings	171
Inspector	172
Adobe Kuler	173
Exporting Your Theme	174
Fireworks Theme Editor	175
Editing Themes	180
Custom Transitions	181
8. Installation and Offline Access	183
Package Definition	183
HTML Manifest	184
Download Process	185
Accessing Online Resources	186
Updating Resources	187
JavaScript Object	188
Events	189
Icon Installation	191
Invitation	191
Icon Name	191
Icon Definition	195
Full Screen	197
Detecting Full Screen	197

Styling the Webapp	199
Mixing It All Together	201
Storing Offline Data	202
9. A Complete Webapp	205
Webapp Structure	205
Offline Manifest	206
Pages	207
Stylesheet	214
Data	214
Script	215
10. Extending the Framework	221
Creating a Plug-in	221
Basic Template	222
Creating Our Plug-in	223
Notable Plug-ins	228
Pagination	228
Bartender	230
DateBox	230
Simple Dialog	232
Action Sheet	234
Plug-ins for Tablets	235
SplitView	235
MultiView	237
Compatible Plug-ins	237
11. Packaging for Stores	239
Store Distribution	240
Custom Distribution	241
Preparing the Package	241
Packaging with PhoneGap	242
PhoneGap Build	243
Index	245

The Mobile Platform

If you are reading this book, you are probably a web designer or a web developer, maybe a jQuery fan or a webapp developer. Before coding, we need to address the mobile ecosystem and where jQuery Mobile fits into it. Let's do it.

Why Do We Need jQuery Mobile?

The first question that you should be asking yourself is *why does jQuery Mobile exist?* Why do we need something special for mobile devices if there are plenty of mobile browsers rendering standard desktop websites?

To answer these questions, let me copy and paste some extracts from my other book *Programming the Mobile Web* (O'Reilly). By the way, I asked for permission from myself for doing that.

Myths of the Mobile Web

As the Web has moved onto mobile devices, developers have told themselves a lot of stories about what this means for their work. While some of those stories are true, others are misleading, confusing, or even dangerous.

It's not the mobile web; it's just the Web!

I've heard this many times in the last few years, and it's true. It's really the same Web. Think about your life. You don't have another email account just for your mobile. (Okay, I know some guys who do, but I don't believe that's typical.)

You read about the latest NBA game on your favorite site, perhaps ESPN; you don't have a desktop news source and a different mobile news source. You really don't want another social network for your mobile; you want to use the same Facebook or Twitter account as the one you used on your desktop. It was painful enough creating your friends list on your desktop, you've already ignored many people...you don't want to have to do all that work again on your mobile.

So, yes...it's the same Web. However, when developing for the mobile web we are targeting very, very different devices. The most obvious difference is the screen size, and yes, that will be our first problem. But there are many other not-so-obvious differences. One issue is that the contexts in which we use our mobile devices are often extremely different from where and how we use our comfortable desktops or even our laptops and netbooks.

Don't get me wrong—this doesn't mean that, as developers, we need to create two, three, or dozens of versions duplicating our work. This is where jQuery Mobile comes to the rescue.

You don't need to do anything special when designing for the mobile web

Almost every smartphone on the market today—for example, the iPhone and Android-based devices—can read and display full desktop websites. Yes, this is true. Users want the same experience on the mobile web as they have on their desktops. Yes, this is also true. Some statistics even indicate that users tend to choose web versions over mobile versions when using a smartphone. However, is this because we really love zooming in and out, scrolling and crawling around for the information we want, or is it because the mobile versions are really awful and don't offer the right user experience? I've seen a lot of mobile sites consisting of nothing but a logo and a couple of text links. My smartphone wants more!

One website should work for all devices (desktop, mobile, TV, etc.)

As we will see, there are techniques that allow us to create only one file but still provide different experiences on a variety of devices, including desktop, mobile, TV, and game consoles. This vision is called “One Web.” Today, there are a lot of mobile devices with very low connection speeds and limited resources—non-smartphones—that, in theory, can read and parse any file, but will not provide the best user experience and will have compatibility and performance problems if we deliver the same document we would for the desktop. Therefore, One Web remains a goal for the future. A little additional work is still required to provide the right user experience for each mobile device, but there are techniques that can be applied to reduce the work required and avoid code and data duplication.

Just create an HTML file with a width of 240 pixels, and you have a mobile website

This is the other fast-food way to think about the mobile web. Today, there are more than 3,000 mobile devices on the market, with almost 50 different browsers (actually, more than 500 different browsers if we separate them by version number). Creating one HTML file as your mobile website will be a very unsuccessful project. In addition, doing so contributes to the belief that the mobile web is not useful.

Mobile Webapps

I'm not going to enter the discussion about mobile web development versus native development. In fact, I believe that the discussion is mostly wrong. Often the discussion is about native code versus JavaScript code or browser apps versus installed apps. However, what these discussions fail to mention is that multiplatform development is quite a challenge in the native development environment because each platform requires a different SDK. Therefore, because the real concerns are ease of development and deployment to multiple mobile devices, mobile web development is a perfect solution for most situations. The term *webapp* has plenty of synonymous or similar concepts, such as mobile webapps, widgets, hybrids, HTML5 apps, and more.

In particular, a mobile webapp differs from typical mobile websites in its purpose. A webapp typically has a more transactional way of thinking in the user interface, emulating native mobile applications. It is still created using web technologies (HTML, CSS, JavaScript, AJAX) but offers an application-similar experience to the user.

Frequently mobile webapps also make use of HTML5 features, such as offline or geolocation access, to provide a better experience. Geolocation is not an official part of the HTML5 specification, but a W3C API of its own; however, it is often mentioned under the HTML5 umbrella.

A webapp can be implemented in many ways (as shown in [Figure 1-1](#)) including:

- Accessed from the browser
- Installed as a full-screen webapp
- As an installed webapp via a package officially implemented by vendors (sometimes called widgets)
- As an installed webapp embedded in a native application, commonly known as hybrid

We will cover how to create these webapps in the rest of the book. For more in-depth information, just cross the street to my other book: [*Programming the Mobile Web*](#) (O'Reilly).

A webapp typically generates new challenges for web designers and developers, such as loading views instead of pages, maintaining a two-way navigation between views, and creating rich controls specifically for touch devices.

So, Again...Why Do We Need jQuery Mobile?

If you read the last few pages (and I'm pretty confident you did), you are aware that mobile web design and development presents new challenges for us. We need to create webapps that are more than simple websites; there are too many devices out there with different browser compatibilities; and there are also too many libraries trying to solve the same problem with mixed community and device support.

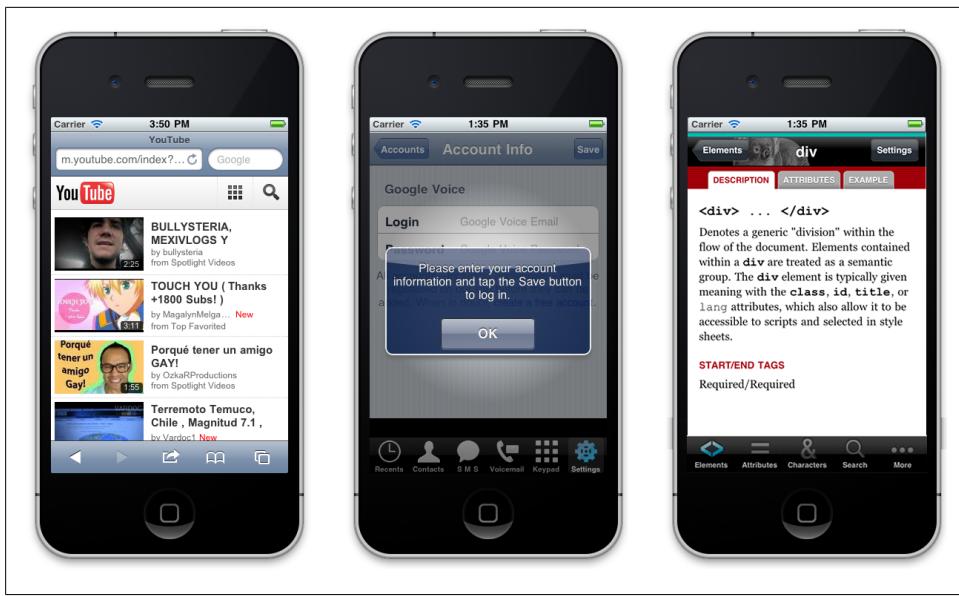


Figure 1-1. A webapp delivered as (from left to right) a browser-based experience, a full-screen installed application, and an embedded webapp inside a native app (hybrid)

That is why jQuery Mobile was created: to help designers and developers create mobile web experiences easily, and for those experiences to be multiplatform, customizable, and with unobtrusive code.

The extensive worldwide jQuery community also provides a great opportunity for the framework's future.

The framework has received official sponsorship and support from many of the biggest companies in this area, including the following:

- Adobe
- Mozilla Corporation
- HP Palm
- BlackBerry/RIM
- Nokia
- DeviceAtlas and dotMobi

What Is jQuery Mobile?

According to an official note at <http://www.jquerymobile.com>:

jQuery Mobile is a unified user interface system across all popular mobile device platforms, built on the rock-solid jQuery and jQuery UI foundation. Its lightweight code is built with progressive enhancement, and has a flexible, easily themeable design.

What jQuery Mobile Is Not

To understand jQuery Mobile, it is very important to know what it is not.

jQuery Mobile is not a jQuery alternative for mobile browsers.

To use jQuery Mobile, you need to include the typical jQuery framework. It is not a replacement; it is a UI layer on top of jQuery.

jQuery Mobile is not a webapp SDK.

You can create the whole mobile experience with jQuery Mobile but you will need some additional work to compile it as native apps. We will see how, why, and when to do this in the next chapters.

jQuery Mobile is not a framework for JavaScript lovers.

Except in the case of certain advanced topics, you won't need any JavaScript code for jQuery Mobile to work. That is great if you are a web designer who hates all of those braces and semicolons.

jQuery Mobile is not the solution for all mobile applications, websites, or games.

However, it covers solutions for most of them. For the others, well...I have to convince you to read my other book somehow.

The Framework

If you don't know what jQuery is, you may be a time traveler from 10 years in the past. If you are Marty McFly, point your browser to <http://jquery.com> and read about this incredibly useful JavaScript framework, the most used one on the Web since 2007.

jQuery Mobile is a framework that delivers webapp experiences to mobile and tablet devices, mainly with touch interfaces, effortlessly, across multiple platforms, and using only HTML5 standard code. A jQuery Mobile app looks like [Figure 1-2](#).

The platform uses the jQuery "core" framework, a JavaScript library, a CSS 3 stylesheet, and some resource images.

jQuery Mobile is comparable to jQuery UI on the desktop side: it's just a UI framework. The name (without any UI inside) leaves you to wonder whether it's a core framework; but I believe the decision was made to take advantage of the power of the jQuery trademark inside the designer and developer world.



The framework was created by the same team as the main jQuery framework, whose leader is John Resig, JavaScript Tool developer for the Mozilla Corporation ([@jeresig](https://twitter.com/@jeresig) on Twitter).

This new platform, like jQuery and jQuery UI, was released as an open source project under a dual license MIT or GPL version 2.

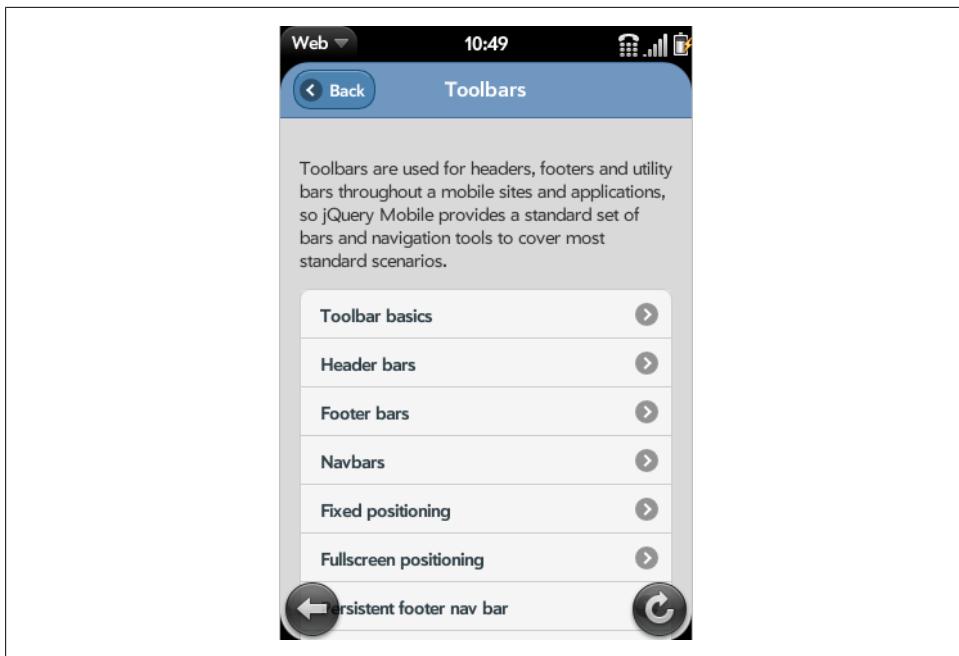


Figure 1-2. A typical jQuery Mobile webapp with standard theming in smartphones, a webOS device in this case



If you want to participate in the development, you can provide patches, fix bugs, participate in the discussion, and work over the active live code on <http://jquerymobile.com/contribute>.

The Mobile and Tablet World

People do not browse the Web only from their desktop. Now we have very different devices with different screen sizes, input mechanisms, and even new features from old friends such as HTML, JavaScript, and CSS.

Mobile devices are here. There is no doubt—there are more than five billion devices worldwide and counting. Tablets are also coming in a big way, with millions in the market.

Device Categories

Right now, we can divide mobile devices into the following categories:

- Mobile phones
- Low-end mobile devices

- Mid- and high-end mobile devices, also known as social devices
- Smartphones
- Tablets

Mobile phones

Okay, we still have mobile phones in some markets. These are phones with call and SMS support. They don't have web browsers or connectivity, and they don't have any installation possibilities. These phones don't really interest us; we can't do anything for them right now.

In a couple of years, because of device recycling and evolving services provided by carriers and manufacturers, such phones will probably not be on the market anymore.

Low-end mobile devices

Low-end mobile devices have a great advantage: they have web support. They typically have a very basic browser, but this is the gross market. Perhaps they aren't the most heavy Internet users today, but this may change quickly with the advent of social networks and Web 2.0 services. If your friends can post pictures from their mobile devices, you'll probably want to do the same, so you may upgrade your phone whenever you can.

Nokia, Motorola, Kyocera, LG, Samsung, and Sony Ericsson have devices for this market. They typically do not have touch support, have limited memory, and include only a very basic camera and a basic music player.

Mid-end/high-end mobile devices

This is the mass-market option for a decent mobile web experience. Mid-end devices maintain the balance between a good user experience and moderate cost. In recent years, this category was also known as social devices, meaning that the users access social sites, such as Facebook or Twitter via the mobile web.

In this category, devices typically offer a medium-sized screen, basic HTML-browser support, sometimes 3G, a decent camera, a music player, games, sometimes touch, and application support. The big difference between these and smartphones is that high-end devices generally are not sold with flat Internet rates. The user can get a flat-rate plan, but he'll have to go out and find it himself. Starting in 2011, many of these devices include WLAN (WiFi), as we can see in [Figure 1-3](#).

Smartphones

There are dozens of smartphone devices on the market, including iPhone, Android-based devices, webOS, Symbian, BlackBerry, and Windows Phone ([Figure 1-4](#)). This is the most difficult category to define. Why aren't some mid-end and high-end devices



Figure 1-3. The Nokia X3-02 Touch and Type: a mid-end touch device, with numeric keypad and WiFi considered “smart” enough to be in this category? The definition of *smart* evolves every year. Even the simplest mobile device on the market today would have been considered very smart 10 years ago.

Typically when you buy a smartphone, you sign up for a one- or two-year contract with a flat-rate data plan. A smartphone, as defined today, has a multitasking identifiable operating system, a modern HTML5 browser, wireless LAN (WLAN, also known as WiFi) and 3G connections, a music player, and several of the following features:

- GPS (Global Positioning System) or A-GPS (Assisted Global Positioning System)
- Digital compass
- Video-capable camera
- TV out
- Bluetooth
- Touch support
- 3D video acceleration
- Accelerometer



Figure 1-4. A sampling of smartphone devices



Some multimedia devices qualify as smartphones for us web creators, but they don't have a phone feature. On this list are the Apple iPod Touch and the Sony PlayStation Portable (PSP). The only difference from tablets is their screen size, less than three inches.

Tablets

A tablet is a device with a large screen (between 6 and 11 inches), a full HTML5 browser, WLAN connection (WiFi), sometimes 3G, touch support, and all the other features that we can find on a smartphone.

In this category, we can find many devices, including the following:

- Apple iPad
- Samsung Galaxy Tab
- BlackBerry PlayBook
- Barnes and Noble Nook Color
- Motorola Xoom
- LG Optimus Pad
- Amazon Fire
- Sony S1 and S2

Operating Systems and Browsers

This book is not intended to delve deeply into the mobile ecosystem. There is a detailed list of operating systems, platforms, and browsers in [Programming the Mobile Web](#)

(O'Reilly). However, if we are going to create mobile web experiences, we need to at least know what we are talking about.

In the mobile world, we can divide the operating systems in two main groups: identifiable operating systems and proprietary ones. In the latter group, we will find mainly phones, low- and mid-end devices.

With the identifiable operating systems, we will be more interested in which OS a device has than its brand and model. I mean, we are not going to develop a webapp for the Samsung Galaxy; we are going to develop a webapp for Android devices. The iPhone may be an exception to this rule, because it is a platform of its own, having as of this writing only one device: the iPhone. (Different versions of the device are just that; for web developers there are no huge differences between an iPhone 4 and an iPhone 3GS.)

Table 1-1 lists the operating systems we can find on today's market in smartphones and tablets.

Table 1-1. Operating systems and browsers available in smartphones, social devices, and tablets

Operating system	Creator	Browser included	Other browsers
iOS	Apple	Safari	Opera Mini and pseudo-browsers
Android	Google	Android Browser	Firefox, Opera Mini, Amazon Silk, Opera Mobile
Symbian	Nokia	Symbian Browser	Opera Mini, Opera Mobile
webOS / Open webOS	HP (formerly Palm)	webOS Browser	
Windows Phone	Microsoft	Internet Explorer	
Windows Mobile	Microsoft	Internet Explorer	Opera Mobile
MeeGo	Nokia	Micro Browser/Nokia Browser	Firefox
BlackBerry OS	RIM	BlackBerry Browser	Opera Mini
Tablet OS	RIM	Tablet OS Browser	
S40	Nokia	Nokia Browser	
Bada	Samsung	Samsung Browser	

Every operating system has different versions, and some allow the user to update to a newer one. Every OS comes with an installed browser, but the user can install and use an alternative browser. Sometimes the manufacturer or the operator from whom the user bought the device installs or replaces the default browser with an alternative, such as Opera Mobile.

If we expand our browser research to low- and mid-end devices, we will find more than 20 other new browsers, including Ovi Browser, NetFront Browser, and Phantom Browser from LG. But that is not the target of jQuery Mobile right now.

What Is a Pseudo-Browser?

A pseudo-browser is a native application that users can install on their devices. They use the same engine as the default browser, but offer extra features over it. There are plenty of examples for iOS, such as SkyFire or Perfect Browser. They all use Safari as the final rendering engine; therefore, for jQuery Mobile, they are not separate browsers.

In *Programming the Mobile Web*, you will find 20 pages with detailed information about browser types and features of each one.

jQuery Mobile Compatibility

jQuery Mobile is a framework intended for touch devices, including smartphones, tablets, and multimedia devices. The compatible list will change with time and as the framework continues to evolve, so it is difficult to publish a complete list here.

The jQuery Mobile 1.0 version is compatible with the following by-default browsers:

iOS

Safari for iPhone, iPod Touch, and iPad from iOS 3.2

Android OS

Android Browser phones and tablets

BlackBerry OS

BlackBerry Browser for smartphones from 5.0 and for tablets

Symbian

Nokia Browser for touch devices

webOS

webOS Browser from webOS 1.4

Bada

Bada Browser

MeeGo

Micro Browser and Nokia Browser (included in Nokia N9)

Windows Phone

Internet Explorer from Windows Phone/Mobile 6.5 and Windows Phone 7.0

Kindle

Browser from Kindle 3

jQuery Mobile is also compatible with the following third-party browsers:

- Opera Mini, fully supported from 5.0 on most devices
- Opera Mobile, fully supported from 10.0 on most devices
- Firefox Mobile

This compatibility list just gives you some information to start with. The compatibility is far more complex than this list because we can cross multiple operating system versions with multiple browser versions with different results. Even newer devices not listed here will be compatible with the library if they support the minimum features that the framework needs.

To simplify: jQuery Mobile will work on every browser with the capabilities to offer the experience that the framework provides. Any modern browser should be included in this list.



Many modern mobile browsers use a WebKit-based engine, like Safari or Chrome for desktop. Any modern WebKit-based mobile browser should be fully compatible with jQuery Mobile. Also Chrome, Firefox, Safari, Opera, and Internet Explorer for desktop are compatible with jQuery Mobile.

Mobile graded browser support

jQuery Mobile uses a table chart to define the compatibility of every device with this library ([Figure 1-5](#)). I would not try to enter in this categorization war if I were you. But you can check it if you want more information at <http://jquerymobile.com/gbs/>.



Many modern desktop browsers, such as Firefox, Google Chrome, Safari, or Internet Explorer, are compatible with jQuery Mobile too. Even if it is not intended for desktop applications, this ability will be useful for testing purposes. However, we will see later that installing an emulation environment will be useful.

I believe the compatibility is far more complex than this table, and for a typical web designer and developer it should be absolutely hidden. There are better ways to know if a feature is available in a mobile browser than trying to categorize each one of them. One solution is right in your hands: *use jQuery Mobile*.

GBS (Graded Browser Support) divides mobile browsers into three categories: A-grade, B-grade, and C-grade. In the jQuery Mobile world, here's what these grades mean:

The screenshot shows a browser window displaying a compatibility matrix for jQuery Mobile. The matrix is organized by platform (rows) and browser/version (columns). The columns represent different browsers and their versions, with grades assigned to each combination. The grades are color-coded: A (green), B (orange), C (red), and N/A (white). The platforms listed are iOS, Symbian S60, Symbian UIQ, Symbian Platform, BlackBerry OS, Android, and Windows Mobile.

Platform	Version	Native		Opera Mobile		Opera Mini		Fennec		Ozone		Netfront		Phonegap	
		8.5	8.65	9.5	10.0	4.0	5.0	1.0	1.1	0.9	4.0	0.9	4.0	0.9	
iOS	v2.2.1	A											A		
	v3.1.3, v3.2	A											A		
	v4.0	A											A		
Symbian S60	v3.1, v3.2	C	C	C		B	C	B		C	C				
	v5.0	A	C	C		A	C	A					A		
Symbian UIQ	v3.0, v3.1			C						C					
	v3.2				C					C					
Symbian Platform	3.0	A													
BlackBerry OS	v4.5	C					C	C							
	v4.6, v4.7	C					C	B					C		
	v5.0	A					C	A					A		
Android	v6.0	A					A						A		
	v1.5, v1.6	A											A		
Windows Mobile	v2.1	A											A		
	v2.2	A			A			C		A			A		
	v6.1	C	C	C	C	B	C	B			C				
	v6.5.1	C	C	C	C	A	C	A							

Figure 1-5. jQuery Mobile maintains a list of browser compatibility on its website

A-grade

A browser with CSS3 media queries capability. These browsers will be fully tested by the jQuery team. However, some features will be automatically disabled if the device does not support them. The framework provides a full experience with AJAX-based animations.

B-grade

Browser has an enhanced experience but without AJAX navigation features.

C-grade

A browser incompatible with jQuery Mobile. This browser will not receive any CSS or JavaScript code from the framework, so the user will see a plain HTML file with the content. We will see later in this book how to handle this situation.



PhoneGap and native development

If you look into the Mobile Graded Browser Support Table of jQuery Mobile, you will find PhoneGap as a browser. PhoneGap is not a browser; it's a framework with which to create hybrids. That is: native applications with a webapp inside. PhoneGap is officially supported by jQuery in many platforms, such as iOS, Symbian, BlackBerry, Android, and webOS.

The great news is that you can use whatever hybrid framework you like; jQuery Mobile will work if it works for PhoneGap. That is because PhoneGap is not a browser by itself; it is just a framework using the native browser engine.

To make it simple: *jQuery Mobile is compatible with the creation of native apps with HTML.*

HTML5 and CSS3

I know most web designers and developers panic about HTML5 and CSS3. Before saying anything about it: don't worry, jQuery Mobile will make everything for you. Therefore, you don't need to know HTML5 or CSS3 to work with this framework. I encourage you to learn them regardless. You will be able to accomplish much more knowing these new standards, but that is for a later discussion.

This book is not intended to teach you HTML5 or CSS3, but it is important to understand some things about them. Many mobile browsers, mainly inside smartphones and tablets, support HTML5, CSS3, and other APIs.

I can talk for hours about HTML5, including its history and what it has to offer to the mobile space.

HTML5 in hard terms is an evolving standard that includes changes to the HTML markup and lot of new APIs in JavaScript (yes, HTML5 is a lot about JavaScript APIs). HTML5 in a more casual term is an umbrella for many modern features inside browsers, including the W3C's formal HTML5 standard, other W3C APIs, CSS3, and nonstandard extensions. You can check <http://mobilehtml5.org> for compatibility information for HTML5 in mobile browsers.

jQuery Mobile uses many of HTML5 features to provide a great and fast experience on mobile browsers. That does not mean that the browser needs to support HTML5 as a whole. In fact, many old browsers still support some HTML5 markup even without knowing its existence. jQuery Mobile uses CSS3 a lot, when possible, for animations, gradients, effects, and UI rendering.

To further arouse your curiosity, with HTML5, CSS3, and other modern techniques, you can provide, with or without a jQuery Mobile experience, the following features:

- Offline access
- Offline storage
- Web sockets
- Geolocation access
- Accelerometer and gyroscope support
- Animations
- 2D and 3D transformations
- Gradients and visual effects
- Viewport management (for zooming support inside the browser)
- Webapp installation metadata
- Integration with native applications
- Multimedia support
- Graphic drawing (vector and bitmap)
- Custom font support

There are several samples and links regarding these features on my blog,
<http://www.mobilexweb.com/>.

Main Features

jQuery Mobile started in August 2010 as a modern framework, including many patterns and best practices for multiplatform development. The main features of the framework are:

- Cross platform, cross device, and cross browser
- UI optimized for touch devices
- Themeable and customizable design
- Usage of nonintrusive semantic HTML5 code only, without the need of any JavaScript, CSS, or API knowledge
- AJAX calls automatically to load dynamic content
- Built on the well-known and supported jQuery core
- Lightweight size, 12Kb compressed
- Progressive enhancement
- Accessibility support

We've already discussed some of these features. Let's deeply analyze others.

Use of Nonintrusive Semantic HTML5

I know you are hungry: you need to see some code. Here you have it. jQuery Mobile creates webapps from standard and semantic HTML5, perfectly suitable for search engine optimization (SEO) and web performance optimization (WPO) purposes:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>My first jQuery Mobile code</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
    <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
    <script type="text/javascript" src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
<div data-role="page" data-theme="a">
    <div data-role="header">
        <h1>jQuery Mobile</h1>
    </div>
    <div data-role="content">

        <ul data-role="listview" data-inset="true" data-divider-theme="b">
            <li data-role="list-divider">Summary</li>
            <li><a href="ch1.html">The Platform</a></li>
            <li><a href="cap2.html">The Page</a></li>
            <li><a href="cap3.html">Lists</a></li>
            <li><a href="cap4.html">Components</a></li>
        </ul>

        <ul data-role="listview" data-inset="true" data-divider-theme="d">
            <li data-role="list-divider">Links</li>
            <li><a href="http://www.mobilexweb.com">Mobile Web Blog</a></li>
            <li><a href="http://www.oreilly.com">O'Reilly Media</a></li>
        </ul>

    </div>
    <div data-role="footer">
        <h4>&copy; 2011 Maximiliano Firtman @firt</h4>
    </div>
</div>
</body>
</html>
```

You can see in [Figure 1-6](#) how this sample renders on many mobile browsers, including non-jQuery Mobile compatible ones, as in [Figure 1-7](#). As you can see, there is no JavaScript code there for initialization or any other stuff. Just some JavaScript includes.

Be patient, we will start analyzing the jQuery code in the following chapters.

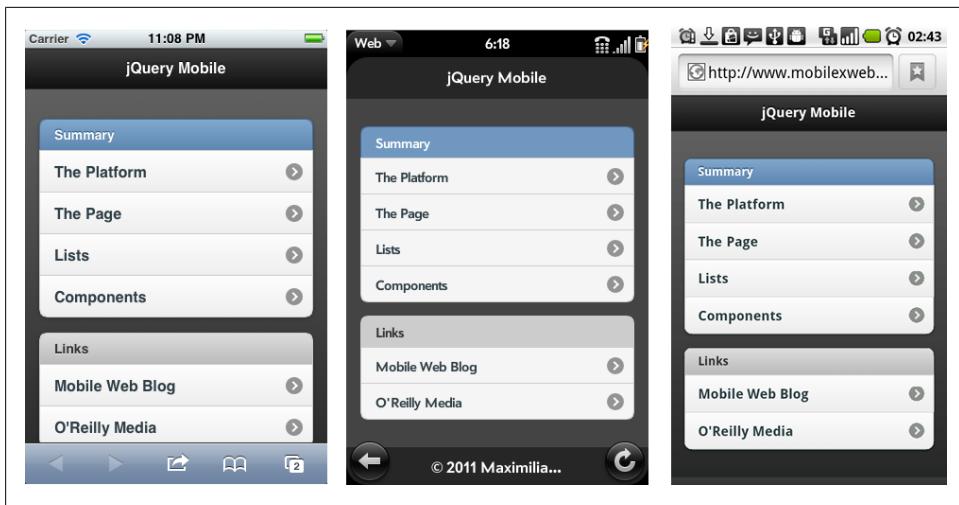


Figure 1-6. How our first simple jQuery Mobile code looks on different devices: iOS, webOS, and Android

jQuery Mobile

- [Summary](#)
- [The Platform](#)
- [The Page](#)
- [Lists](#)
- [Components](#)

- [Links](#)
- [Mobile Web Blog](#)
- [O'Reilly Media](#)

© 2011 Maximiliano Firtman @firt

Figure 1-7. On noncompatible browsers, jQuery Mobile falls back to a simple, fully functional HTML file

Progressive Enhancement

Progressive enhancement is a simple but very powerful technique used in web design that defines layers of compatibility that allow any user to access the basic content, services, and functionality of a website, while providing an enhanced experience for browsers with better support of standards. jQuery Mobile is totally built using this technique.

The term *progressive enhancement* was coined by Steven Champeon in 2003 (<http://www.hesketh.com>), and while this approach wasn't defined for the mobile web specifically, it is perfect for mobile web design.

Progressive enhancement has the following core principles:

- Basic content is accessible to all browsers.
- Basic functionality is accessible to all browsers.
- Semantic markup contains all content.
- Enhanced layout is provided by externally linked CSS.
- Enhanced behavior is provided by unobtrusive, externally linked JavaScript.
- End user browser preferences are respected.

This list sounds like jQuery Mobile's feature list, doesn't it? That's right. A jQuery Mobile application will also work on a very basic browser without CSS or JavaScript support. And that is a great feature for a mobile webapp.

Accessibility Support

From Wikipedia:

Web accessibility refers to the inclusive practice of making websites usable by people of all abilities and disabilities. When sites are correctly designed, developed, and edited, all users can have equal access to information and functionality.

Web accessibility inside mobile browsers has just begun; however, jQuery Mobile is now fully compatible with W3C's WAI-ARIA specification on compatible browsers (<http://www.w3.org/TR/wai-aria/>). At the time of this writing, only iOS 4.0 or higher is compatible with this specification with the feature called VoiceOver.

Therefore, a jQuery Mobile webapp will provide an accessible experience to users with visual disabilities on iPhone, iPod, and iPad.

Testing Webapps

We have already mentioned that a jQuery Mobile webapp will work on almost every modern desktop browser. However, it would be better if we could test them in a more accurate environment (see [Figure 1-8](#)).

To test the mobile webapp in different environments, we can use:

- Real devices
- Remote labs
- Emulators
- Simulators
- Lot of friends

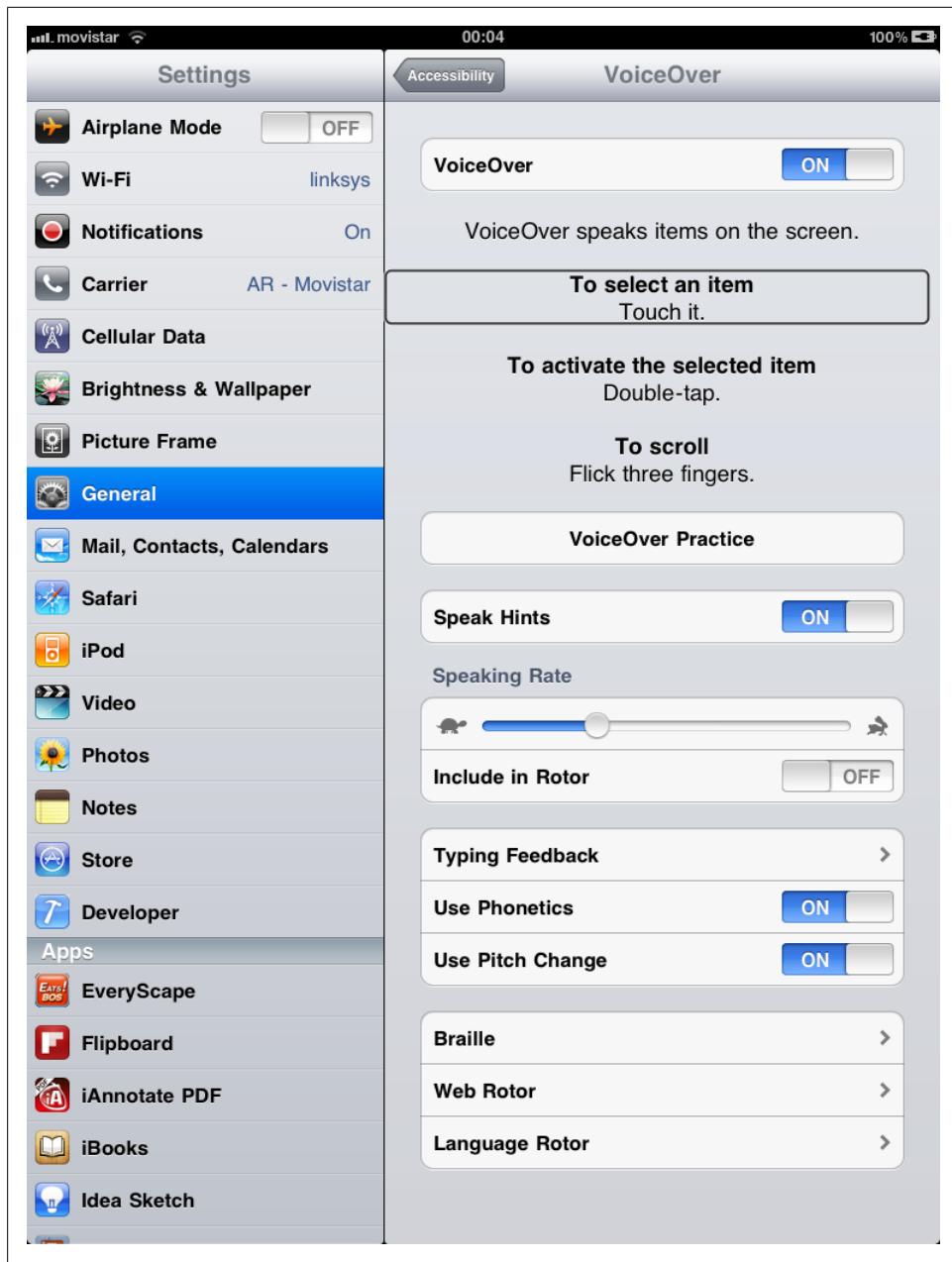


Figure 1-8. Testing jQuery Mobile accessibility from an iPad

Emulators and Simulators

The most useful tools for our work will be emulators and simulators. Generally speaking, an emulator is a piece of software that translates compiled code from an original architecture to the platform where it is running. It allows us to run an operating system and applications on another operating system. In the mobile development world, an emulator is a desktop application that emulates mobile device hardware and operating systems, allowing us to test and debug our applications and see how they are working. The browser, and even the operating system, is not aware that it is running on an emulator, so we can execute the same code as on the real device.

We should also add to our mobile development environment classic tools for project and configuration management, such as bug tracking, version control, and project management tools. [Figure 1-8](#) shows how you can test jQuery Mobile accessibility from an iPhone, iPod, or iPad with iOS 4.0 or higher. Go to Settings → General → Accessibility and activate VoiceOver. Now close your eyes and browse your website using your fingers and ears.

[Figure 1-9](#) shows how an Android Emulator provides a full Android OS on your desktop with images for different devices including tablets, such as Galaxy Tab or Nook Color.

Emulators are created by manufacturers and offered to developers for free, either standalone or bundled with the Software Development Kit (SDK) for native development.

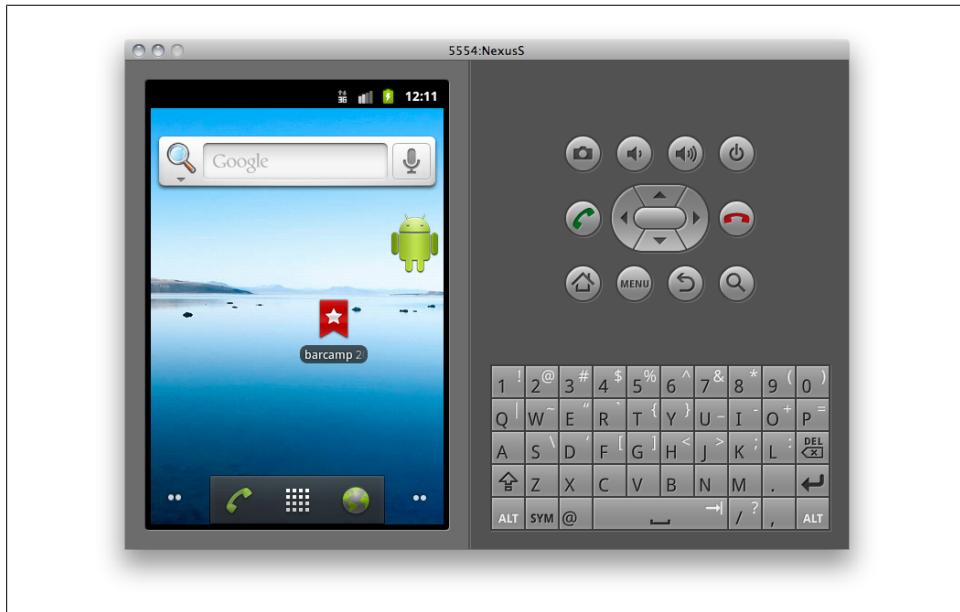


Figure 1-9. An Android Emulator

There are also operating system emulators that don't represent any real device hardware but rather the operating system as a whole. These exist for Windows Mobile and Android.

On the other hand, a simulator is a less complex application that simulates some of the behavior of a device, but does not emulate hardware and does not work over the real operating system. These tools are simpler and less useful than emulators. A simulator may be created by the device manufacturer or by some other company offering a simulation environment for developers. In mobile browsing, there are simulators with pixel-level simulation, and others that neither create a skin over a typical desktop browser (e.g., Firefox or Safari) with real typography nor render engine simulation. [Figure 1-10](#) shows how the iOS Simulator provides you with an iPad for free inside your Mac. The same can happen for other tablets, also with Windows or Linux desktop machines.



Figure 1-10. The iOS simulator

Even with emulators, the final rendering and performance will not be exactly the same as in the real device. Therefore, real device testing is a good practice, even if we are going to do it only on some key devices.

For mobile web development, we will find emulators from Nokia, Symbian, BlackBerry, Android, webOS, and Windows Mobile and simulators from Apple for the iPhone and iPad (though only for Mac OS X).

Some browser-based emulators (that finally work on many different platforms), like the Opera Mobile emulator, are also available.

Table 1-2 shows the available emulators and simulators for download.

Table 1-2. Available mobile and tablet emulators and simulators for download

Name	Platform	Type	Browsers available	Windows	Mac	Linux
iOS Simulator	iOS	Simulator	Safari	No	Yes	No
Android Emulator	Android	Emulator	Android Browser and downloadable	Yes	Yes	Yes
HP webOS Emulator	webOS	Emulator	webOS Browser	Yes	Yes	Yes
Nokia Symbian Emulators	Symbian	Emulator	Internal Browser and downloadable	Yes	No	No
Windows Phone Emulator	Windows Phone	Emulator	Internet Explorer	Yes	No	No
Nokia Series 40 Emulators	Nokia OS	Emulator	S40, Ovi Browser, Opera Mini	Yes	No	No
BlackBerry Simulators	BlackBerry OS	Emulator	BB Browser, downloadable	Yes	No	No
BlackBerry PlayBook Simulator	Tablet OS	Emulator	Internal Browser	Yes	Yes	Yes
Opera Mobile Emulator	Many	Browser Emulator	Opera Mobile	Yes	Yes	Yes
Opera Mini Simulator	Many	Online Browser Emulator	Opera Mini	Yes	Yes	Yes
PhoneGap Simulator	Many	Simulator	PhoneGap hybrid	Yes	Yes	Yes
Adobe Device Central	Many	Simulator	Many	Yes	Yes	No

An up-to-date list of emulator download URLs can be found at <http://www.mobilexweb.com/emulators>.

Remote Labs

A remote lab is a web service that allows us to use a real device remotely without being physically in the same place. It is a simple but very powerful solution that gives us access to thousands of real devices, connected to real networks all over the world, with a single click. You can think of it as a remote desktop for mobile phones.

The most useful services on the market are the following:

Keynote DeviceAnywhere (commercial)

<http://www.deviceanywhere.com>

Perfecto Mobile (commercial)

<http://www.perfectomobile.com>

Nokia Remote Device Access for Symbian and MeeGo (free)

<http://www.mobilexweb.com/go/rda>

Samsung Lab.Dev for Android (free)

<http://www.mobilexweb.com/go/labdev>

For updated information on this topic, go to <http://www.mobilexweb.com/go/labs>.

O'Reilly Ebooks—Your bookshelf on your devices!



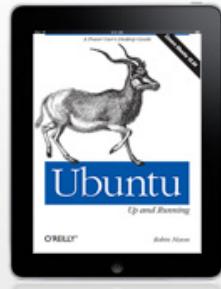
PDF



ePub



Mobi



APK



DAISY

When you buy an ebook through oreilly.com you get lifetime access to the book, and whenever possible we provide it to you in five, DRM-free file formats—PDF, .epub, Kindle-compatible .mobi, Android .apk, and DAISY—that you can use on the devices of your choice. Our ebook files are fully searchable, and you can cut-and-paste and print them. We also alert you when we've updated the files with corrections and additions.

Learn more at ebooks.oreilly.com

You can also purchase O'Reilly ebooks through the iBookstore, the [Android Marketplace](http://Android.Marketplace), and Amazon.com.

O'REILLY®

Spreading the knowledge of innovators

oreilly.com