jQuery Mobile

DEVELOP AND DESIGN

Kris Hadlock

jQuery Mobile

Develop and Design

Kris Hadlock

jQuery Mobile: Develop and Design
Kris Hadlock

Peachpit Press

1249 Eighth Street
Berkeley, CA 94710
510/524-2178
510/524-2221 (fax)

Find us on the Web at: www.peachpit.com
To report errors, please send a note to: errata@peachpit.com
Peachpit Press is a division of Pearson Education.
Copyright © 2012 by Kris Hadlock

Acquisitions Editor: Michael Nolan
Project Editor: Rebecca Gulick
Development Editor: Robyn G. Thomas
Contributing Writer: Jay Blanchard
Technical Reviewer: Jay Blanchard
Production Coordinator: Myrna Vladic
Compositor: David Van Ness
Copyeditor: Gretchen Dykstra
Proofreader: Patricia Pane
Indexer: Valerie Haynes-Perry
Cover Design: Aren Howell Straiger
Interior Design: Mimi Heft

Notice of Rights
All rights reserved. No part of this book may be reproduced or transmitted in any form by any means electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission o the publisher. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Notice of Liability
The information in this book is distributed on an "As Is" basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Peachpit Press shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

Trademarks
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 Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

ISBN-13: 978-0-321-82041-9
ISBN-10: 0-321-82041-X

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

To my wife, Lisa, who carried our first child while I wrote this book.
Only true love can withstand the amount of time that it
takes to write a book while having a new baby.
And to my son, Lucas, words cannot
express the love I feel for you.

Acknowledgments

There are many people I would like to thank for the opportunity and help they gave before, during, an after this book was being written: Neil Salkind, for helping me navigate the world of publishing and for his support while I was writing. Robyn Thomas for her patience. Jay Blanchard for stepping in when needed and providing excellent technical reviews. Rebecca Gulick for helping to move things along. Michael Nolan for working out the details. All my customers, for understanding how busy I've been. And, of course, Peachpit for giving me the opportunity to write for you.

About the Author

Kris Hadlock has been a web developer and designer since 1996, working on projects for companies such as SPIN Magazine, IKEA, United Airlines, JP Morgan Chase, Canon, and Phoenix Children's Hospital, to name a few. Kris is a featured columnist and writer for numerous websites and design magazines, including Peachpit.com, InformIT.com, IBM developerWorks (*www.ibm.com/developerworks*), and *Practical Web Design* magazine. His other books include *Ajax for Web Application Developers* and *The ActionScript 3.0 Migration Guide*. He is the founder and lea developer-designer of Studio Sedition (www.studiosedition.com), specializing in the fusion of form and function.

Contents

Smartphone, tablet, and e-reader statistics are showing an unprecedented adoption rate, making the mobile web a very hot topic and requiring a new set of skills from web developers and designers. Mobile device usage is skyrocketing; according to Nielsen's third-quarter 2011 Mobile Media Repor "44 percent of U.S. mobile subscribers now own a smartphone device, compared to 18 percent just tv years ago." That's more than double in two years, and "the number of smartphone subscribers using the mobile Internet has grown 45 percent since 2010." As for tablets, in June 2011 AMI-Partners (Access Markets International) forecasted that "tablet adoption among businesses with between 1 and 1,000 employees will grow by 1,000 percent by 2015."

Let's not forget e-readers, which are becoming very affordable and are more advanced then ever, increasing in shipment volume, as "year-over-year growth was 167%" according to International Dat Corporation (IDC). With the introduction of the latest Kindle, mobile Internet access is now becomin a normal experience.

With these increases in adoption rate, there will no doubt be high demand for web developers who ca create rich mobile web experiences. The jQuery Mobile framework gives web developers a quick and easy way to create mobile web experiences, making the mobile web space hard to ignore.

Why jQuery Mobile?

As a web developer, you don't have to use the jQuery Mobile framework to create a mobile web experience. So why use it? For starters, the framework is built on the highly respected and widely use jQuery core and jQuery user interface (UI) foundation. It's currently sponsored by companies such as Mozilla, Palm, Adobe, Nokia, BlackBerry, and more. Plus, it works seamlessly across all popular mobile device platforms. The jQuery Mobile team is actively and regularly offering new releases, blogging about new features, and keeping their comprehensive online documentation up to date.

Most web developers and designers agree that browser and cross-platform testing is something they would rather not spend their time on. Imagine all of the devices that could potentially be accessing your mobile website. Then imagine having to test all of those platforms each and every time you buil a mobile website—this would be painstaking and incredibly time-consuming. jQuery Mobile gives you this support from the start, as the team prides the framework on its approach to supporting a wid variety of mobile platforms. The framework is built on clean, semantic HTML, which ensures compatibility with a majority of web-enabled devices.

The framework also includes accessibility features, such as WAI-ARIA (Web Accessibility Initiative Accessible Rich Internet Applications), a technical specification published by the World Wide Web Consortium regarding the increase of accessibility of webpages, which are integrated into the framework to support screen readers, such as VoiceOver on Apple iOS and other assistive technologies. Simply including the jQuery Mobile framework in your website unobtrusively transforms your code from semantic HTML into a rich, interactive, and accessible mobile experience using jQuery and CSS. As you'll see throughout this book, the jQuery Mobile approach makes mobil web development incredibly easy, quick, and efficient, leaving the platform and browser testing up to the jQuery Mobile team.

jQuery Mobile isn't exclusively for web developers; web designers have access to the jQuery UI, which provides complete design control over mobile web applications. Built-in UI widgets, such as list views, dialogs, toolbars, search mechanisms, and a full set of form elements, are all customizable

via the theme framework. Later in the book, you'll also learn about ThemeRoller, which lets you create up to 26 theme swatches using a simple, web-based interface. User experience (UX) designers also get some love, with access to stencils for OmniGraffle and Visio. And, of course, if you want to get geeky with it, the application programming interface (API) is available to web developers. As a web developer, you can configure defaults, handle many different events, and work with several exposed methods and properties.

An emerging community is helping to support the framework with a number of third-party apps and frameworks that you can use to build jQuery Mobile apps. In addition, jQuery Mobile compatible plug-ins and extensions are popping up to help web developers integrate custom widgets and add capabilities to the existing core functionality.

One very important third-party framework is blurring the line between native and mobile web-based development. As an HTML5 app platform, PhoneGap allows you to author native applications using web technologies. With PhoneGap, your web applications can easily be ported into native apps that can do things like retrieve contact information, access cameras, use geolocation, store data, and mucl more. To learn more about PhoneGap, visit phonegap.com. There's even a section in the jQuery Mobile online documentation about it. With these sorts of possibilities, you no longer need to progra multiple versions of a native application. This makes native application development less desirable, because the same application you develop for an iPhone would need to be completely redeveloped fo Android, BlackBerry, Windows Mobile, and others. Oh, and don't forget that every new release will need to be updated for every one of these different platforms. The jQuery Mobile framework provide mobile web experiences that rival native application development by giving you instant access to we applications and websites via the web browser, eliminating the need to download and install mobile applications.

Who this Book is For

This book is for people who have basic HTML experience and are interested in creating mobile websites using the jQuery Mobile framework.

Who this Book is not For

This book is not for people who have never created a webpage.

How you will Learn

In this book, you'll learn by doing. Each chapter includes sample code and descriptions to give you a deep understanding of how things work. You can also find the code samples on the book's website (www.peachpit.com/jquerymobile).

What you will learn

This book will teach you everything from the basics of how to create pages to custom-theming them and developing your own jQuery Mobile content-management system with WordPress and Drupal. B the time you finish this book, you'll be a jQuery Mobile expert.

Wrapping Up

The jQuery Mobile framework is a powerful framework that is supported by mobile industry leaders. It can easily be added to an existing website to create a mobile web experience that is not only touch-friendly, but also supported on a majority of the leading mobile platforms as well as handicap accessible. Design control, page transitions, widget integration, scripting, API access, and much mor are all at your fingertips through this framework's easy-to-use features and built-in progressive

enhancement techniques.

Supported jQuery Mobile Platforms

The jQuery Mobile framework supports the majority of modern desktop, smartphone, tablet, and e-reader platforms. Rather than spending your time testing multiple devices, you can rest assured that you're offering support for many platforms from the start. This is because the jQuery Mobile platfor takes a progressive enhancement approach, which not only brings rich interactive experiences to devices, but also provides support for older browsers and phones. When a browser fails to recognize certain HTML5-specific code, the webpage renders as a simple, yet functional webpage. Users on older phones or browsers are familiar with a limited web experience, therefore this approach still renders an acceptable basic HTML webpage in these cases.

Currently, Android and Apple iOS are the leading mobile operating systems. Android has the largest operating system market share (44.2 percent), while Apple has the largest smartphone market share i the United States (28.6 percent). The Windows, BlackBerry, SymbianOS, and Palm/HP webOS operating systems comprise the remaining majority of smartphone market share (**Figure 1**). All of these platforms/operating systems are supported by the jQuery Mobile framework.

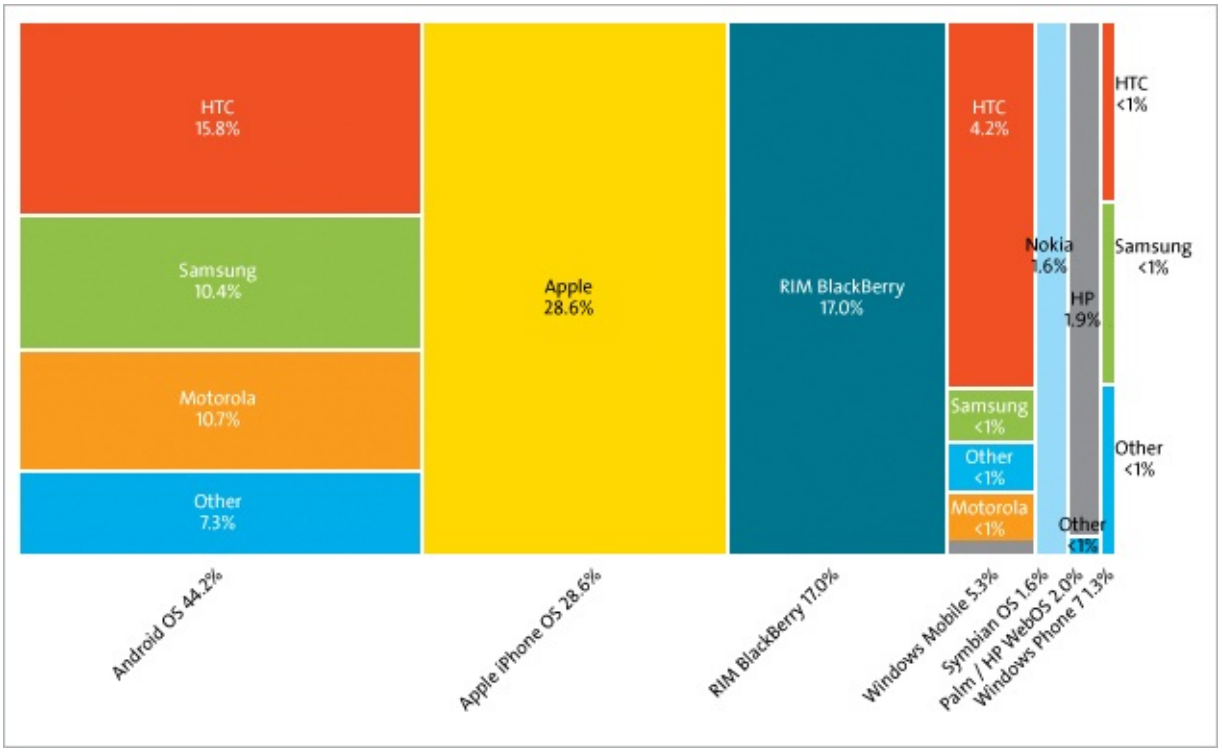

Figure 1. Operating system market share as of 2011.

Major Platforms

The following list provides a bit more information on the major platforms that are fully supported by the jQuery Mobile framework.

IOS

iOS is Apple's mobile operating system. Originally developed for the iPhone, it has been extende to support other Apple devices such as the iPod Touch, iPad, and Apple TV. iOS uses Safari as its

web browser.

ANDROID

Android is an open-source software project and operating system led by Google. It has been used in a plethora of phones, tablets, and other devices since its release under the Apache license. Android uses Google Chrome as its web browser.

WINDOWS PHONE

Windows Phone is Microsoft's mobile operating system. The system is integrated with third-part and other Microsoft services. Windows Phone uses Internet Explorer Mobile as its web browser.

BLACKBERRY

BlackBerry is powered by a proprietary mobile operating system, offered on its own set of smart-phones and tablets.

WEBOS

webOS is the open-source operating system used by Palm devices. webOS uses the WebKit layou engine in its web browser, simply named Web.

Graded Support

jQuery Mobile uses a three-tier, graded list of the platforms that are supported by the framework. Th tiers are A, B, and C. A includes a full experience with the option of Ajax-based page transitions, B includes the same experience minus the Ajax-based page transitions, and C is a basic, yet functional, HTML experience.

A-grade support includes all the major mobile operating systems that were mentioned previously and more, including Apple iOS, Android, Windows Phone, BlackBerry, Palm/HP webOS, Kindle 3, Kindl Fire, and more. B-grade support includes BlackBerry 5, Opera Mini, and Nokia SymbianOS. C-grade support includes BlackBerry 4, Windows Mobile, and all older smart-phones and feature phones. Thi list is always evolving; to see an up-to-date list, visit jquerymobile.com and check out the supported

platforms.

Part I: The Foundation of jQuery Mobile

1. Understanding jQuery

The jQuery framework is the backbone of the jQuery Mobile framework, so it's helpful to know som of the fundamentals of the jQuery framework before developing jQuery Mobile websites. Although it's not required, understanding jQuery will make using jQuery Mobile even easier than it already is, especially if you're interested in writing any sort of custom functionality.

jQuery is a robust yet lightweight JavaScript library that simplifies JavaScript coding and extends th capabilities of Cascading Style Sheets (CSS). In addition, it eliminates cross-browser compatibility issues and is CSS3 compliant. This means quicker scripting, less testing, and less coding for the different ways that browsers handle certain functionality. The jQuery framework is truly an example of how web development should be: You get what you expect the first time, every time.

Getting Started

To get started with jQuery, you first need to download the framework from jquery.com and include it in your webpage, or you can simply reference it via the Google, Microsoft, or jQuery content delivery network (CDN). I recommend and most often reference the library via a CDN because it's faster. A CDN will distribute your content across multiple, geographically dispersed servers, so the user receives the closest available file. Plus, Google and Microsoft both offer versions that support secure socket layers (SSL) via HTTPS, must-haves if you're doing any sort of development under SSL. To include the library via a CDN (we'll use Google as an example), use the script element to include it within the `<head>` elements or at the end of your webpage.

Click here to view code image

```
<script type="text/javascript"
    src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js">
</script>
```

Including JavaScript files within the `<head>` elements is the traditional approach. However, according to Yahoo!, "80 percent of the end-user response time is on the front-end." Most of that time is spent downloading assets, such as style sheets, images, scripts, and so on. It's obviously important to reduc the number of assets, but it's also becoming more common to include JavaScript at the end of an HTML file. This is because scripts block parallel downloads, meaning that other assets will not download until each script is downloaded individually. To ensure that you're placing your scripts in the correct place, simply include them before the closing `</html>` tag.

It's also best to use the minified version in production environments, because it's smaller than the source version. In addition, although the packed version is smaller than the minified version, it requires client-side processing time to decompress the file and it's not available in the most recent versions. According to Yahoo!, "In a survey of ten top U.S. websites, minification achieved a 21 percent size reduction."

jQuery Fundamentals

If you're familiar with writing JavaScript and CSS, then writing your first jQuery script will feel ver familiar, yet maybe slightly odd. The jQuery framework is a JavaScript library, meaning that it's bui with JavaScript. The fundamentals are the same, as you're essentially still writing JavaScript, it just so happens that you'll be writing in a way that uses the jQuery framework. In other words, while ther may be added enhancements to certain fundamentals, the core of JavaScript—variables, functions,

conditional statements, and so on—has not changed. So, you'll still be using the `var` keyword, `if` and `switch` statements, and functions, but you'll surely notice a lot of additional enhancements and different ways of writing things, namely accessing HTML elements.

jQuery offers many enhancements to the JavaScript language and, as mentioned, the best part is that it's cross-browser compatible, so you don't have to worry about writing multiple versions of the sam script to handle different browsers anymore, which is quite a relief. This is especially useful when working with events, Ajax (discussed later in this chapter), and other functionality that traditionally requires some conditional statements to determine how the browser will interpret the code.

Selecting HTML elements

Rather than continually using `document.getElementById()` to access HTML elements within your webpage, you can simply use a jQuery selector `jquery()` or the `$()` function, which is the shorthand version and the one that I will be using throughout this book. Using the jQuery selector not only give you fewer characters to type, but also lets you do far more than access elements by `id`. With jQuery selectors, you can also access an array of HTML elements or access an element or object by name. The jQuery selector wraps an element or set of elements into a jQuery object, allowing you to apply jQuery methods to the object itself. And that's not all: You can even access elements by class name c use CSS selectors such as `:first-child`, `:nth-child`, among many others. Here are a few examples:

- To access an element by ID using the jQuery selector, use the `#id` selector, just as you would wit CSS when attaching a class to an element by ID:

```
$('#foo');
```

- To access an element by class name, use the `.class` selector, just as you would when creating a CSS class. If you have multiple elements with the same class name, they will all be selected.

```
$('.foo');
```

- To target a specific element with a class name, add the element name before the class name:

```
$('div.foo');
```

- You can even target a specific element or set of elements using an element selector:

```
$('div');
```

If you're familiar with CSS, you're probably starting to see the pattern unfold and the endless possibilities. The options available for structuring a CSS class are the same ones you can use to acces an HTML element with the jQuery selector. This is how jQuery enhances JavaScript and CSS, taking the two and merging their syntax into a new language that enhances the overall user experience, whe properly used, of course.

Managing events and functions

Events are used by jQuery to react to user interactions on your webpage, such as mouse clicks. Event are very easy to manage in jQuery, and they're reliable across all the major browsers, which is a big deal, because this isn't always the case with traditional JavaScript. jQuery gives you the ability to intercept many different events from any existing HTML element. Typically, events are used to perform some sort of function. In the following example, a `click` event is bound to a `div` element wit `foo` as the class name:

Click here to view code image

```
$('div.foo').click(function(e) {
    // Your custom code here
});
```

By binding the `click` event to the `div`, a function is associated with clicking the mouse on that particular `div`. Therefore, anytime `div.foo` is clicked, your custom code in the handler function will b executed.

If you're familiar with traditional JavaScript, this syntax probably looks strange. Don't worry—it dic to me as well when I first started using it, but once I caught on, I liked it. Once you're familiar with i you'll see that this code syntax is well contained, easy to understand, and easy to write. Let's break down the previous example:

1. The jQuery selector is used to select `div.foo`, which then becomes a jQuery object.
2. The `div.foo` jQuery object then uses the `click` method to fire a handler function when `div.foo` clicked.
3. The handler function is used to execute the custom code. The handler also has access to the `eventObject`, which is passed as an argument. In this example, it is the `e` argument in the handle function.

If you use the jQuery selector to select a class name that's being used by multiple HTML elements and assign a mouse event, that mouse event will be bound to all of those elements automatically. Therefore, in the example above, if you had multiple `div` elements with `foo` as the class name, the `click` event would be bound to all of them. When working with events that are bound to multiple elements, it's important to consider scope to ensure that any custom code you're executing is applied to the desired element.

Click here to view code image

```
$('div.foo').click(function(e) {
    alert($(this).html());
});
```

In this example, `$(this)` is used to access the current element in scope.

Note

Scope is the enclosing context that values are associated with. In this case, the function is the enclosing context and `$(this)` is the value.

`$(this)` is the same as using the `this` keyword in JavaScript; in this case, it applies to the element tha is currently firing the `click` event. When considering scope, `this` always refers to the object associate with the enclosing context. In the example, `div.foo` is the object and the handler function for the `clic` event is the enclosing context. jQuery will automatically assign the element that is clicked as `$(this)` inside your anonymous handler function. This lets you access the element that executed the event, even when the same code is applied to multiple elements. Functions are used to execute a script or a set of scripts as the result of an event or a simple and direct call to the function.

As you'll learn later in this book, the `bind` method is often used by the jQuery Mobile framework to handle custom events.

The custom events provided by jQuery Mobile create useful hooks for development beyond the nativ functionality of the framework, meaning that you can intercept certain existing events and add your

own custom code. Such events include `touch`, `mouse`, and `window` events, all of which you'll learn how to extend later in this book. The syntax for the `bind` method in jQuery is very similar to the previous code example.

Click here to view code image

```
$('div.foo').bind('click', function(e) {
    // Your custom code here
});
```

The difference is that you can react to multiple events with the same handler:

Click here to view code image

```
$('div.foo').bind('mouseenter mouseleave', function(e) {
    // Your custom code here
});
```

Note

The difference between a function and a method is that a function is stand-alone and a method is associated with an object. In this case, the `bind` method is associated with the jQuery object.

Or you can handle custom events, such as those implemented by the jQuery Mobile framework:

Click here to view code image

```
$('div.foo').bind('displayMyName', function(e, myName) {
    alert('My name is: '+ myName);
});
$('button').click(function () {
    $('div.foo').trigger('displayMyName', ['John Smith']);
});
```

Binding events will be covered in greater depth when we discuss how to bind jQuery Mobile events with custom handler functions.

Waiting for documents to be ready

A common practice among developers is to wait for a webpage to load before executing any JavaScript. The reason for doing this is to ensure that the webpage elements are available before trying to access or manipulate them. Attempting to access unavailable document elements can lead to unexpected behavior and potentially break all your subsequent JavaScript. With traditional JavaScript the most common way to wait for the page to load is to use the `window.onload` event. However, this approach happens after the document loads, because it actually waits for all images and banners to load within a webpage, which can delay your scripts tremendously. Luckily, jQuery provides a `ready` event that lets your code respond immediately when the document becomes available.

Click here to view code image

```
$(document).ready(function() {
    // Your custom code here
});
```

First, you need to select the document object itself and then apply the `ready` event. When the `ready` event fires, it will execute a function that will contain your custom code. Continuing with the previou

examples, you need to add the custom code to the `ready` event to ensure that the document is ready before trying to apply the `click` event:

Click here to view code image

```
$(document).ready(function() {
    $('div.foo').click(function() {
        alert($(this).html());
    });
});
```

Or you can use a shortcut version, which eliminates the need to access the `document` and set the `ready` event.

Click here to view code image

```
$(function() {
    $('div.foo').click(function() {
        alert($(this).html());
    });
});
```

Using the `ready` event is the de facto standard with any jQuery development. Not only does this approach ensure that your code will fire at the appropriate time, it also lends itself to creating completely unobtrusive code, meaning that you can write all your jQuery in a separate, external file that is simply referenced from your HTML webpage.

Applying special effects

jQuery is well-known for the special effects it lets you create without using third-party plug-ins, such as Flash. The library provides many techniques for incorporating animation into a webpage. Animations can create a more visually appealing webpage, or they can serve other, more practical purposes, such as providing visual feedback to user interactions. Many prebuilt animation methods a included in the jQuery framework, such as `fadeIn`, `fadeOut`, `fadeTo`, `show`, `toggle`, `slideUp`, and `slideDown`, among others.

```
$(document).ready(function() {
    $('div.foo').fadeIn();
});
```

Although these prebuilt animation methods appear to be independent, many of them are powered internally by the `animate` method. The `animate` method can be used to animate any numerical CSS property for an HTML element, such as `height`, `opacity`, `left`, and `right`.

Click here to view code image

```
$(document).ready(function() {
    $('div.foo').click(function() {
        $(this).animate(function(){
            height: '+=50'
    }, 1000, function() {
            // Your custom code, when the animation is complete
        });
    });
});
```

In this example, when `div.foo` is clicked, its `height` property is increased by 50 pixels over a time period of 1 second (1000 milliseconds), and when the animation is complete, a callback function will execute your custom code. A callback function is used to delay the execution of code until something

sample content of jQuery Mobile: Develop and Design

- read Here For Now: Living Well With Cancer Through Mindfulness
- read The Violets of March pdf
- read Cruel Death online
- click Principles of Economics (7th Edition) pdf, azw (kindle)
- read online Yellow Blue Tibia
- read online Master and Commander (Aubrey & Maturin Series, Book 1)

- http://studystrategically.com/freebooks/Shadow-s-Witness--Sembia--Gateway-to-the-Realms--Book-2-.pdf
- http://yachtwebsitedemo.com/books/The-Violets-of-March.pdf
- http://serazard.com/lib/Cruel-Death.pdf
- http://metromekanik.com/ebooks/Routledge-Philosophy-Guidebook-to-Mill-On-Liberty--Routledge-Philosophy-Guidebooks-.pdf
- http://sidenoter.com/?ebooks/Yellow-Blue-Tibia.pdf
- http://www.shreesaiexport.com/library/The-Well-Spoken-Thesaurus.pdf