



Learn by doing: less theory, more results

jQuery 2.0 Animation Techniques

Bring your websites to life with animations using jQuery

Beginner's Guide

Adam Culpepper

Dan Wellman

[PACKT] open source 
PUBLISHING community experience distilled

jQuery 2.0 Animation Techniques Beginner's Guide

Bring your websites to life with animations using jQuery

Adam Culpepper

Dan Wellman

[PACKT] open source 
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

jQuery 2.0 Animation Techniques Beginner's Guide

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: December 2013

Production Reference: 1171213

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.
ISBN 978-1-78216-964-2

www.packtpub.com

Cover Image by Jeremy Segal (info@jsegalphoto.com)

Credits

Authors

Adam Culpepper
Dan Wellman

Reviewers

Jeff Byrnes
Emil Lerch
Doug Sparling

Acquisition Editors

Harsha Bharwani
Akram Hussain

Lead Technical Editors

Mayur Hule
Larissa Pinto

Technical Editors

Iram Malik
Menza Mathew
Rahul U. Nair
Pratish Soman

Project Coordinator

Anugya Khurana

Proofreaders

Lauren Harkins
Elinor Perry-Smith

Indexer

Mehreen Deshmukh

Production Coordinator

Shantanu Zagade

Cover Work

Shantanu Zagade

About the Authors

Adam Culpepper is a frontend web developer who works for Envoc, a software development firm, located in Baton Rouge, Louisiana. He is very active in his community; organizing, founding, and sitting on the board of many industry-related groups and organizations in his area. When he's not reading, writing, working, organizing, or attending a community event, he's spending time with his great friends and family or his girlfriend Amber, his son Aiden and his cat Emma. Adam can be reached at his website (www.adamculpepper.net) or through Twitter (@adamculpepper).

I dedicate this book to my girlfriend, Amber, who tolerates my time spent reading, writing, and learning.

To my son Aiden and my cat Emma, who both frequently break my concentration while writing.

To my mom, who has always encouraged me to reach for my dreams (except for my childhood dream of growing up to be a puppy).

To the best damn boss in the world, Calvin Fabre.

Also, I'd like to give special thanks to the period, comma, and the semicolon. Without you guys, I would have gotten a lot more restful nights.

Dan Wellman is an author and web developer based on the South coast of the UK. By day he works alongside some of the most talented people he has had the pleasure of calling colleagues, for a small, yet accomplished digital agency called Design Haus. By night he writes books and tutorials on a range of frontend topics. He is hopelessly addicted to jQuery. His life is enriched by four wonderful children, a beautiful wife, and a close circle of family and friends. This is his fifth book.

I would like to thank the hugely supportive and patient editorial team at Packt, without whom this book would not exist. I would also like to thank the reviewers, especially Ben Nadel and Cyril Pierron, who put aside their own personal projects and dedicated countless hours to ensuring the book's technical accuracy. I'd also like to say a big Hey! to some of my closest friends, in no particular order; Andrew Herman, Steev Bishop, Aaron Matheson, Eamon O'Donoghue, James Zabiela, Mike Woodford, and John Adams.

About the Reviewers

Jeff Byrnes is a software engineer with over 8 years of experience working in web development and operations. Educated as a musician at Berklee College of Music, self-taught as an engineer, he has experience ranges from frontend to backend and systems. HTML5, CSS3, JS, PHP, Git and version control, Puppet, Chef, Vagrant, Bash, Linux, deployment, automation, and analytics; these are all the technologies and skills with which he is experienced and proficient.

Currently, part of the Operations team at EverTrue, Jeff spends his days supporting the engineering team, providing automation, server monitoring and maintenance, workflow improvements, and deployment solutions, as well as providing help desk support to the company at large.

I'd like to thank Dale Cruse for connecting me with the opportunity to assist with this book's publication, and my better half Leah for supporting me as I worked on it.

Emil Lerch is a technical architect, leading teams to build web-based systems for enterprises since 1995. He lives in Portland, Oregon with his wife Kelly and children Kathryn and Jack. His blog is <http://emilsblog.lerch.org> and he can be reached on twitter @elerch.

My lovely wife and children for giving me the time to help review the book.

Doug Sparling works as a web and mobile software developer for Andrews McMeel Universal, a publishing and syndication company in Kansas City, MO. As long-time employee of the company, he has built everything from the GoComics Android app to registration, ecommerce systems, web services, and various web sites using Ruby on Rails. He's now busy building another site in Rails and porting a Perl-based e-mail system to Go. Some of the AMU properties include GoComics.com, PuzzleSociety.com, Doonesbury.com, and Dilbert.com.

He is also the directory of technology for the small web development firm called New Age Graphics (newage-graphics.com). After creating a custom CMS using C# and ASP.NET, all work has moved to WordPress since WordPress 3.0 was released, eliminating the need to ever run Windows again.

Doug is the author of a popular jQuery plugin, **jClock**.

Doug is a passionate advocate for WordPress and has written several WordPress plugins, can be found on the WordPress.org forums answering questions (and writing sample code) under the username "scriptrunner", and occasionally plays grammar nerd as a volunteer on the WordPress Codex Documentation team.

Other experience includes PHP, JavaScript, jQuery, Erlang, Python, Magento, and Perl. Doug was also the co-author for a Perl book (Instant Perl modules) and is a reviewer for other Packt books, including Mastering Android 3D Game Development and WordPress Web Application Development, as well as The Well Ground Rubyist, 2nd Edition and Learn Android in a Month of Lunches for Manning Publications.

In his less than ample spare time, Doug enjoys spending time with his family. Other passions include photography, writing music, hitting drums and cymbals with sticks, playing briscola, meditation, and watching countless reruns of Firefly, Buffy the Vampire Slayer, and Doctor Who.

Many thanks to Packt for giving me the opportunity to review a book on jQuery.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- ◆ Fully searchable across every book published by Packt
- ◆ Copy and paste, print and bookmark content
- ◆ On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Getting Started	7
Animation on the Web	7
The power of animated UIs	9
When to use animations	9
When not to use animations	10
Animation checklist	10
Animating with jQuery	11
Creating the project folder	12
The template file	12
A basic animation example	15
Time for action – creating an animated loader	15
Summary	19
Chapter 2: Image Animation	21
Image animation	21
Fading animations	22
Configuring the animations with arguments	22
Time for action – setting up the markup and styling	24
Scripting the image slider	26
Time for action – scripting the image rotator	26
Time for action – extending the pause-on-hover functionality	29
Time for action – extending the previous and next link features	30
Time for action – extending the pagination functionality	33
Summary	37

Chapter 3: Background Animation	39
Background-color animation	39
Introducing the animate method	39
Time for action – animating the body background-color	40
Chaining together jQuery methods	41
The illusion of depth with parallax	42
Time for action – creating the stage and adding the styling	43
Time for action – animating the background position	45
Automatic background animation	46
Time for action – creating an automatic background animation	46
Let's make it diagonal!	48
Time for action – animating the background diagonally	48
Parallax background on page elements	50
Time for action – setting up the markup and styling	50
Time for action – scripting our parallax script	51
Summary	54
Chapter 4: Navigation Animation	55
Creating simple navigation animations	55
Configuring addClass() and removeClass()	56
Time for action – setting up our navigation	56
Using the stop() method	58
Time for action – adding the stop() method	58
Animating the window with scrollTop()	58
Time for action – scripting our smooth scrolling animation	59
Smooth scrolling and page background color	61
Time for action – creating the super animation	62
Summary	65
Chapter 5: Form and Input Animation	67
Using simple form animations	68
Time for action – creating the form	68
Time for action – adding our animation styles to the form	70
Form validation animations	71
Time for action – basic dialog form validation	71
Time for action – animating form validation errors	73
Summary	75
Chapter 6: Extended Animations with jQuery UI	77
Obtaining and setting up jQuery UI	78
Creating a new template file	79
The new effects added by jQuery UI	80
Using the effect API	80

The bounce effect	81
Syntax	81
Configuration options	81
Time for action – using the bounce effect	82
The highlight effect	84
Syntax	84
Configuration options	84
Time for action – highlighting elements	84
The pulsate effect	86
Syntax	86
Configuration options	87
Time for action – making an element pulsate	87
The shake effect	89
Syntax	89
Configuration options	89
Time for action – shaking an element	90
The size effect	92
Syntax	92
Configuration options	92
Time for action – resizing elements	92
The transfer effect	95
Syntax	95
Configuration options	95
Time for action – transferring the outline of one element to another	95
Using effects with the show and hide logic	98
The blind effect	98
Syntax	99
Configuration options	99
Time for action – using the blind effect	99
The clip effect	101
Syntax	101
Configuration options	101
Time for action – clipping an element in and out	101
The drop effect	103
Syntax	103
Configuration options	103
Time for action – using the effect	104
The explode effect	106
Syntax	106
Configuration options	106
Time for action – exploding an element	107
The fold effect	108
Syntax	108
Configuration options	108

Time for action – folding an element away	108
The puff effect	110
Syntax	110
Configuration options	110
Time for action – making an element disappear in a puff	110
The slide effect	113
Syntax	113
Configuration options	113
Time for action – sliding elements in and out of view	113
The scale effect	115
Syntax	115
Configuration options	115
Time for action – scaling an element	116
Easing functions	118
Time for action – adding easing to effects	119
Using an object literal to add easing	120
Animating between different colors	121
Time for action – animating between colors	121
Class transitions	122
Time for action – transitioning between classes	122
Summary	124
Chapter 7: Custom Animations	127
The animate method	127
Per-property easing	128
An alternative syntax for animate()	128
Animating an element's position	129
Time for action – creating an animated content viewer	130
Time for action – initializing variables and prepping the widget	132
Time for action – defining a post-animation callback	134
Time for action – adding event handlers for the UI elements	136
Skinning the widget	138
Time for action – adding a new skin	138
Animating an element's size	141
Time for action – creating the underlying page and basic styling	142
Time for action – defining the full and small sizes of the images	143
Time for action – creating the overlay images	144
Time for action – creating the overlay wrappers	145
Time for action – maintaining the overlay positions	147
Creating a jQuery animation plugin	150
Time for action – creating a test page and adding some styling	150
Creating the plugin	153
Time for action – adding a license and defining configurable options	153

Time for action – adding our plugin method to the jQuery namespace	154
Time for action – creating the UI	156
Time for action – creating the transition overlay	158
Time for action – defining the transitions	159
Using the plugin	161
Summary	165
Chapter 8: Other Popular Animations	167
Understanding Proximity animations	167
Time for action – creating and styling the page	168
Time for action – preparing the page for sliding functionality	171
Time for action – animating the scroller	174
Time for action – adding the mouse events	175
Time for action – adding keyboard events	176
Animating page headers	178
Time for action – creating an animated header	178
Animating text using the marquee effect	180
Time for action – creating and styling the underlying page	180
Time for action – retrieving and processing the post list	183
Time for action – animating the post links	189
Summary	191
Chapter 9: CSS3 Animations	193
CSS3 2D transforms	194
Understanding the matrix	194
Translate	195
Scale	196
Skew	198
Rotation	200
Working with transforms	201
Using jQuery and transforms	202
CSS3 3D transforms	203
Animated rotation with jQuery and CSS3	204
Time for action – animating an element's rotation	204
Problems with IE	208
Animated skewing	209
Time for action – creating the underlying markup and basic styling	210
Time for action – initializing the widget	212
Time for action – animating an element's skew	216
Time for action – skewing an element from left to right	221
Time for action – wiring up the controls	224
Summary	226

Chapter 10: Canvas Animations	227
Learning the HTMLCanvasElement interface	228
Using the canvas element	228
Understanding context methods	228
Native shapes	229
Drawing using a path	230
Drawing images and patterns	231
Text strings	232
Applying transformation methods	232
Pixel manipulation	233
Drawing to the canvas	234
Time for action – drawing to the canvas	234
Animating the canvas	239
Time for action – creating an animation on the canvas	239
Time for action – animating the white crosses	241
Time for action – animating the red crosses	243
Creating a canvas game	247
Time for action – creating the initial page	248
Time for action – creating the initial script	249
Time for action – adding the aliens to the page	250
Time for action – moving the aliens	252
Time for action – adding handlers to control the ship	254
Summary	258
Appendix: Pop Quiz Answers	259
Chapter 1, Getting Started	259
Chapter 2, Image Animation	259
Chapter 3, Background Animation	260
Chapter 4, Navigation Animation	260
Chapter 5, Form and Input Animation	260
Chapter 6, Extended Animations with jQuery UI	261
Chapter 7, Custom Animations	261
Chapter 8, Other Popular Animations	262
Chapter 9, CSS3 Animations	262
Chapter 10, Canvas Animations	262
Index	265

Preface

jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML, and is the most popular JavaScript library in use today. Using the features offered by jQuery, developers are able to create dynamic web pages. This book will act as a resource for you to create animation and advanced special effects in your web applications, by following the easy-to-understand steps mentioned in it.

jQuery 2.0 Animation Techniques Beginner's Guide will allow you to master animation in jQuery to produce slick and attractive interfaces that respond to your visitors' interactions. You will learn everything you need to know about creating engaging and effective web page animations using jQuery. The book uses many examples and explains how to create animations using an easy, step-by-step, beginner's guide approach.

This book provides various examples that gradually build up the reader's knowledge and practical experience in using the jQuery API to create stunning animations. The book starts off by explaining how animations make your user interface interactive and attractive. It explains the various methods used to make the element being animated appear or disappear. It provides a set of steps to create simple animations and show fading animations.

You can later learn how to make complex animations by chaining different effects together as well as how to halt a currently running animation. You will find out how to slide your animation elements and learn to create custom animations that can be complex and specialized.

You will learn how to obtain and set up jQuery UI—the official user interface library for jQuery. This book will tell you how to animate a page's background image, and will teach you how to make images scroll in a certain direction and at a certain speed depending on the movement of the mouse pointer.

What this book covers

Chapter 1, Getting Started, covers the basics including downloading jQuery and setting up a development area, a brief history of animation on the Web, when and where not to use animation, how animation can enhance an interface, and the animation methods exposed by jQuery. A basic example of animation is also covered.

Chapter 2, Image Animation, uses a simple approach to creating an image slider. We then build features into the slider and are left with a script worthy of your next development project.

Chapter 3, Background Animation, takes us through the journey of creating animated background images and background color when our user scrolls down our site. This very subtle animation adds a lot of aesthetic appeal to websites.

Chapter 4, Navigation Animation, covers creative ways to add animation to navigation on our websites. We'll be fading the background color of our web page and smooth scrolling to clicked links on the page.

Chapter 5, Form and Input Animation, focuses on animation that is triggered by our user's interaction with the form. We will guide our users through our form using animations for form validation and to give our form a better experience overall.

Chapter 6, Extending Animations with jQuery UI, looks at the additional effects added by jQuery UI, the official UI library built on top of jQuery. We look at each of the 14 effects as well as covering the easing functions built into the library.

Chapter 7, Custom Animation, focuses on the `animate()` method, which jQuery provides for us as a means of creating custom animations not already predefined. This extremely powerful method allows us to animate almost any CSS-style property to easily create complex and attractive animations.

Chapter 8, Other Popular Animations, looks at some common types of animations found on the web including proximity animations triggered by the mouse pointer, animated headers, and a modern-day equivalent to the marquee element.

Chapter 9, CSS3 Animations, covers how we can use CSS3 to create attractive animations driven by the latest CSS transforms and how jQuery can be used to make the process easier.

Chapter 10, Canvas Animations, looks at the HTML5 canvas element and shows how it can be used to create stunning animations without the use of Flash or other proprietary technologies. The book closes with an in-depth example teaching how to create an interactive game using nothing but HTML and JavaScript.

What you need for this book

To get the most out of this book you should have some knowledge of front-end development, preferably including JavaScript. Experience with jQuery is also preferable, but is not essential as all techniques used in the book are discussed in full.

You should have a computer capable of running the latest browsers and preferably an Internet connection. A code editing development software package will be of help, but again is not essential provided you have a text editor of some kind.

Who this book is for

This book is written for web designers and front-end developers who already have good knowledge of HTML and CSS. While not required, some experience with jQuery or JavaScript is helpful. If you want to learn how to animate the user interface of your web applications with jQuery, then this book is for you.

Conventions

In this book, you will find several headings appearing frequently.

To give clear instructions of how to complete a procedure or task, we use:

Time for action – heading

1. Action 1
2. Action 2
3. Action 3

Instructions often need some extra explanation so that they make sense, so they are followed with:

What just happened?

This heading explains the working of tasks or instructions that you have just completed.

You will also find some other learning aids in the book, including:

Pop quiz – heading

These are short multiple-choice questions intended to help you test your own understanding.

Have a go hero – heading

These practical challenges give you ideas for experimenting with what you have learned.

You will also find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "The `fadeIn()` and `fadeOut()` methods perform the least complex animations available via jQuery"

A block of code is set as follows:

```
$("#next").click(function(event) {  
    activeSlide++;  
    rotate();  
    event.preventDefault();  
});
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
$("#slider, #prev, #next").hover(function() {  
    clearInterval(timer);  
    pause = true;  
}, function() {  
    timer = setInterval(rotate, speed);  
    pause = false;  
});
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "In this case, we clear the whole canvas, removing the space ship and any surviving aliens, and print the text **GAME OVER!** to the center of the canvas."



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title through the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website, or added to any list of existing errata, under the Errata section of that title.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Getting Started

Welcome to the jQuery 2.0 Animation Techniques Beginner's Guide. Over the course of the book we'll look at each and every method that produces or controls animations available in the jQuery JavaScript library. We'll see how the methods are used, the arguments they are able to accept, and the different behavior they produce. We'll also look at how to use a range of accompanying resources, including selected jQuery plugins and the jQuery UI library.

In this introductory chapter, we'll look at the following topics:

- ◆ A brief history of animation on the Web
- ◆ Why animating your UIs is important
- ◆ Animation methods provided by jQuery
- ◆ The template file used by each of the examples
- ◆ A basic animation example

Animation on the Web

In 1989, CompuServe released GIF89a, an enhanced version of the popular GIF image format, which allowed a sequence of frames to be stored as a single image and played by supporting software.

The GIF format was already popular on what passed for the Internet in those days (remember, the World Wide Web didn't even exist until 1991) due to its small file size, lossless compression, and wide support. The enhanced version which allowed animations that anyone could create themselves, provided they had supporting software, quickly became popular.

In addition to animated GIFs, browser vendors added support for proprietary HTML elements that handled animation natively, such as the `<blink>` and `<marquee>` elements, which added different animated effects to text.

Neither of these elements was particularly attractive or successful, and the W3C, as well as leading industry accessibility and usability experts advised against their use in most cases. Different browsers at that time supported one or the other of these elements, but not both. Both elements were added by their respective vendors as part of the original browser wars.

In the late 1990s, popular browsers added support for a technique known as **Dynamic HTML (DHTML)**, which allowed scripting languages to modify the contents of a page after the page had loaded. DHTML wasn't any single technology, but rather a collection of techniques (JavaScript, CSS, DOM, and so on) that worked together to enable a basic level of interactivity and/or animation.

In fact, DHTML made it possible to create quite advanced animations, but restrictions in the early implementations of the required technologies, as well as hugely varying browser support made DHTML tricky at best.

This era also saw the release and the rise of Flash (and Shockwave, a competing technology that was eventually subsumed by Macromedia—which was later acquired by Adobe in 2005), a vector and raster graphics format that allowed audio and video streaming, frame-by-frame animation, and a host of other features. Flash quickly became popular, and at the time of writing is still the number one format for web-based video, browser-based gaming, and advertising.

Gradual standardization of the DOM across (most) browsers, as well as the rise of JavaScript libraries such as jQuery, which abstracted away the differences that remained between browsers, have opened up animation to a much wider range of people than ever before. The term DHTML isn't often used these days because of its connotations with poor support between browsers, but the underlying principles and techniques that drive many interactive and animated sites remain similar.

Today, in addition to the animations made plausible and accessible by JavaScript libraries, we have much newer and much more exciting possibilities with CSS3 and native HTML elements such as the `<canvas>` element, which provides complete pixel-level control over an area of the page. We'll be looking at some CSS3 animation techniques, as well as the `<canvas>` element in more detail towards the end of the book. Flash-based animation is on the decline for the first time this century, and new technologies are poised on the horizon.

The power of animated UIs

Modern operating systems use animations constantly to engage their users and to create a more compelling computing experience. When used in the right way, animations provide assistance to the users of the system, to lead and guide them through different tasks, provide context or feedback, and reinforce positive actions.

A good example of this is the way that applications are minimized in Windows 7 or OS X—the application appears to squish down into the icon on the taskbar/dock, which shows the user where to go when they want to return to the application. It's the simple details like this that can be most effective.

Good animations can lend an air of sleek professionalism to an interface and make it appear more advanced or more modern. Apple's iPhone (or iPad) is a perfect example—the seamless use of subtle animations and transitions within the operating system and its applications allow the user to connect with the device in a profoundly satisfying and immersive way. Anything that appears or disappears is faded smoothly in or out, and menus and content panels slide in or out from the top or the sides. Sudden events can unsettle or distract users, but a well-timed animation can help to make them aware that something is happening or something is about to happen.

Be warned though, badly executed, clumsy, or overly pointless animations can do the opposite, making your interface appear basic, poorly designed, or inferior. No animation can be better than poor animation. Even if your application works perfectly, superfluous animations can leave your users feeling frustrated and cause them to forgo your application or website.

Desktop computers and a rapidly growing number of mobile and handheld devices are easily powerful enough to handle quite complex animations, and with integrated hardware acceleration and more refined CSS3 and HTML5 making its way into the latest browsers, the possibilities of what can be achieved on the Web are increasing exponentially.

When to use animations

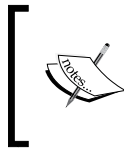
Animations can make a great impression and enhance the user experience in the following situations:

- ◆ When showing or hiding windows, pop ups, and content panels
- ◆ When something is moved to a different area of the window or page
- ◆ When something has changed state on the page as a result of the action of the user
- ◆ To lead the user to a specific call to action or bring their attention to something important

When not to use animations

Too many animations in unnecessary places can be damaging. Try and avoid animations, or at least give them serious consideration, in the following situations:

- ◆ When an action needs to be repeated very frequently by the user
- ◆ Where the devices known to use the system are likely to be incapable of displaying the animation adequately
- ◆ On time-sensitive actions or processes



Bear in mind that these are guidelines only, not laws which must be obeyed at all costs, and they are certainly not definitive. There are few situations where animations should never, ever be used and few situations where they must always be used.

Use your judgment to determine whether an animation is suitable for your application or page and its intended audience. If possible, give your users the chance to enable or disable animations based on their own personal preferences.

Animation checklist

Before implementing an animation in our pages or applications, consider the following checklist of questions:

- ◆ Is the animation appropriate for your target users?
- ◆ Is the animation practical?
- ◆ Does the animation add value or enhance the user experience?
- ◆ Will the animation run at appropriate speeds on the devices that are most likely to be used?

If you can answer yes to all of the above questions, the animation will probably be a positive feature. If you answered no to any of these questions, you probably need to stop and think about what you are trying to achieve by adding the animation, and whether or not it could be better achieved in some other manner.

Animating with jQuery

jQuery (<http://jquery.com>) provides a range of animation methods natively, without the use of additional effect libraries or plugins. There are, however, many plugins contributed from the online community, including jQuery UI (<http://jqueryui.com>), the official UI library for jQuery, which extends jQuery's animation capabilities. Natively, jQuery provides methods that add sliding and fading behavior with minimal configuration and which work cross-browser. It also exposes methods related to managing the animation queue, and provides a means for creating custom animations that work on almost all numerical CSS styles. Over the course of this book, we'll look at every animation method that the library contains in detail. These methods are listed here with descriptions of each:

Methods	Description
<code>animate()</code>	It performs a custom animation of a set of CSS properties.
<code>clearQueue()</code>	It removes from the queue all items that have not yet been run.
<code>delay()</code>	It sets a timer to delay execution of subsequent items in the queue.
<code>dequeue()</code>	It executes the next function on the queue for the matched elements.
<code>fadeIn()</code>	It displays the matched elements by fading them to opaque.
<code>fadeOut()</code>	It hides the matched elements by fading them to transparent.
<code>fadeTo()</code>	It adjusts the opacity of the matched elements.
<code>fadeToggle()</code>	It displays or hides the matched elements by animating their opacity.
<code>finish()</code>	It stops the currently-running animation, removes all of the queued animations, and completes all of the animations for the matched elements.
<code>hide()</code>	It hides the matched elements.
<code>queue()</code>	It shows the queue of functions to be executed on the matched elements.
<code>show()</code>	It displays the matched elements.
<code>slideDown()</code>	It displays the matched elements with a sliding motion.
<code>slideToggle()</code>	It displays or hides the matched elements with a sliding motion.
<code>slideUp()</code>	It hides the matched elements with a sliding motion.
<code>stop()</code>	It stops the currently-running animation on the matched elements.
<code>toggle()</code>	It displays or hides the matched elements.

It's important to note that there are two properties that can change the Global jQuery Object. These are listed below:

Property	Description
<code>jQuery.fx.interval</code>	It is the rate (in milliseconds) at which animations fire.
<code>jQuery.fx.off</code>	It globally disables all animations.

All in all, it gives us a powerful and robust environment to easily add almost any type of animation that we can conceive.

Animation is also a popular theme for plugins, with many available plugins that bring different types of animations to our fingertips for instant implementation with minimal configuration. We'll look at several plugins later in the book.

Creating the project folder

So, that's the template file that we'll be referring to and using in the code examples throughout the book. Let's also take a moment to look at the folder structure that the example files use. Create a project folder and call it `jquery-animation` or any similar name. Within this, create three new folders and call them `css`, `img`, and `js`.

The HTML pages we create will go into the `jquery-animation` folder alongside the subfolders. All of the CSS files we create will go into the `css` folder, and all of the images that we use in the examples will go into the `img` folder. The jQuery library and any additional script files we use or create will go into the `js` folder. This is also the directory structure you'll find if you download and unpack the accompanying code archive containing all of the examples.

The template file

Each of the example files we'll create throughout the course of this book will rely on a common set of elements. Rather than repeatedly showing these same elements in every single code section and examples in the book, we'll take a look at them just once here:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title></title>
    <link rel="stylesheet" href="css/.css">
  </head>
  <body>
```

```
<script src="js/jquery.js"></script>
<script>
  $(function() {

    });
  </script>
</body>
</html>
```

**Downloading the example code**

You can download the example code files for all Packt books you have purchased from your account at <http://packtpub.com>. If you purchased this book elsewhere, you can visit <http://packtpub.com/support> and register to have the files e-mailed directly to you.

Save a copy of this file into the `jquery-animation` folder we just created and call it `template.html`. This is the base file that we'll use for every single example, so when we start working through the examples and I say *add the following markup to the <body> of the template file*, it means insert it directly between the opening `<body>` tag and the first `<script>` tag in the template file we just created in the preceding code. Whenever we add any JavaScript to the template file, it will be added within the anonymous function in the second `<script>` tag.

Let's take a look at what the template file contains. We start out with the HTML5 doctype declaration, as we'll be using plenty of HTML5 elements in our examples. We also set the `lang` attribute of the `<html>` element to `en` and the `<meta>` tag's `charset` attribute to `utf-8`, neither of which are strictly required, but are nevertheless best practice.

Next comes an empty `<title>` element, to which we can add the name of each example and a `<link>` element with an incomplete `href`, ready for us to add the name of the stylesheet that each example will use.

Since the versions prior to **Internet Explorer 9 (IE9)** don't support any HTML5 elements, we need to use Remy Sharp's `html5shiv` script to make this browser use HTML5 correctly. We can link to the online version of this file for convenience using a conditional comment that targets all of the versions of IE lower than version 9. Feel free to download `html5.js` and store it locally if you plan on playing with the examples in IE while disconnected from the Internet.

To get the most out of the examples throughout the book, it would probably be wise to upgrade to the latest stable release versions of the most common browsers, which at the time of writing are *Firefox 24*, *Chrome 30*, *Safari 6*, and *Opera 17* (although expect these to change quite rapidly).



It's important to note that jQuery 2.0 doesn't support *oldIE*, meaning IE8 and below. For this reason, we won't be covering any browser compatibility fixes for those versions of IE.

If your project needs to be compatible with IE8 or older, you'll need to use **jQuery 1.10** or below. Furthermore, if your project uses HTML5 elements and needs to be compatible with IE8 or below, you'll need to use `html5shiv` (<https://code.google.com/p/html5shiv>).

IE9 does support a lot of HTML5 and CSS3, so using the `html5shiv` file in general will only be required as long as IE8 holds its top spot as the world's most used browser. At the time of writing, Internet Explorer 8 had a market share of 21 percent worldwide according to NetMarketShare (<http://netmarketshare.com>). With 19 percent, IE10 comes in second and Chrome 29, Firefox 23 and IE9 are following shortly behind. The `<body>` tag of the page is empty, except for some `<script>` tags. We'll obviously use jQuery in every example, so the first tag links to that. The current version of jQuery is 2.0 at the time of writing (but like the browser versions, this is likely to change pretty quickly).

Throughout the book, we'll be using a local version of jQuery, so that we won't have to rely on being connected to the Internet or worry about Internet slowness. However, in most cases, in production, it's recommended to link to one of the 4 CDNs (Content Delivery Network) for jQuery. These can be found below:

CDN hosted by	URL
jQuery	http://code.jquery.com
Google	https://developers.google.com/speed/libraries/devguide?csw=1#jquery
Microsoft	http://asp.net/ajaxlibrary/cdn.ashx#jQuery_Releases_on_the_CDN_0
CDNJS	http://cdnjs.com/libraries/jquery

In the second `<script>` tag we have an empty function, into which all of the example JavaScript code we write will go. We pass the jQuery object into our anonymous function and alias it to the `$` character. Although not strictly necessary (except in the example where we create a jQuery plugin), this is another good habit to get into.

A basic animation example

Let's look at a basic example of the kind of animation that can help reassure our visitors that something is happening. If the user performs an action, and the result is not displayed immediately, providing feedback to the user that their action is in the process of being executed is a helpful use of animation.

In the next screenshot we can see the loading indicator centered beneath the **Initiate the action** button. It features three separate loading bars which sequentially light up to show that something is happening. Each bar is styled slightly differently.

Time for action – creating an animated loader

In this example we'll create a simple animated loading indicator that we can start when a particular process is initiated and stop once the process has completed.

1. Open up the template file that we just looked at and add the following `<button>` element to `<body>` (this should go before the `<script>` elements):
2. Next, in the empty function in the second `<script>` element at the bottom of the page, add the following code:

```
<button id="go">Initiate the action</button>

var loader = $("<div></div>", {
    id: "loader"
}).css("display", "none");
var bar = $("<span></span>").css("opacity", 0.2);
var loadingInterval = null;
for (var x = 0; x < 3; x++) {
    bar.clone().addClass("bar-" + x).appendTo(loader);
}
loader.insertAfter("#go");

function runLoader() {
    var firstBar = loader.children(":first"),
        secondBar = loader.children().eq(1),
        thirdBar = loader.children(":last");

    firstBar.fadeTo("fast", 1, function () {
        firstBar.fadeTo("fast", 0.2, function () {
            secondBar.fadeTo("fast", 1, function () {
```