

Making Everything Easier!™

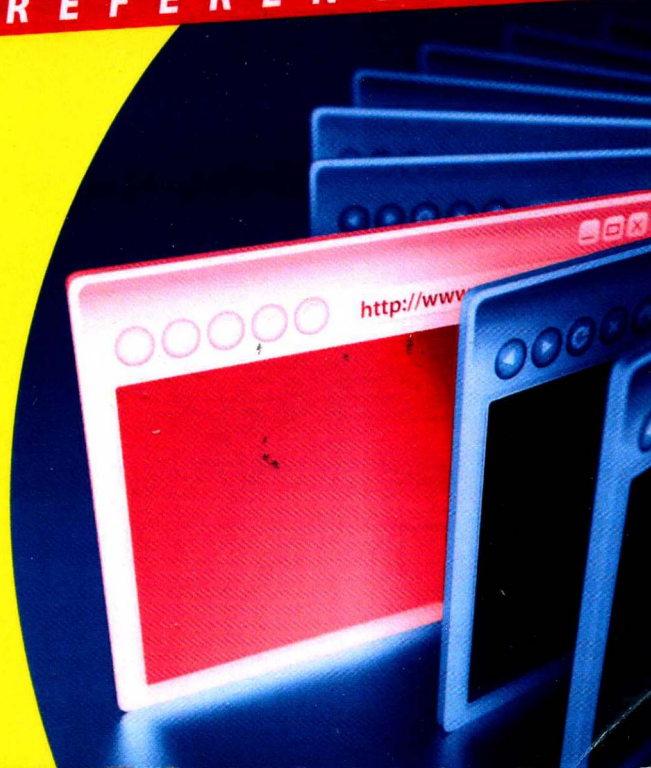
HTML5

FOR
DUMMIES®

Q U I C K R E F E R E N C E

Learn to:

- Create rich Internet applications using HTML5
- Build mobile sites with video
- Manage the HTML5 database capabilities
- Test, debug, and deploy your sites



Andy Harris

*Author of HTML, XHTML, and CSS
All in One For Dummies*

HTML5 FOR **DUMMIES®** **QUICK REFERENCE**

by Andy Harris




WILEY

Wiley Publishing, Inc.

HTML5 For Dummies® Quick Reference

Published by
Wiley Publishing, Inc.
111 River Street
Hoboken, NJ 07030-5774

www.wiley.com

Copyright © 2011 by Wiley Publishing, Inc., Indianapolis, Indiana

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Trademarks: Wiley, the Wiley Publishing logo, For Dummies, the Dummies Man logo, A Reference for the Rest of Us!, The Dummies Way, Dummies Daily, The Fun and Easy Way, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

For technical support, please visit www.wiley.com/techsupport.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2011924127

ISBN: 978-1-118-01252-9

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1



WILEY

About the Author

Andy Harris once owned a TRS-80 Model I. It's still in the garage. He remembers fondly typing BASIC code into that machine and wondering how it really worked. He eventually taught himself enough programming to work as a consultant while pursuing a career in special education. He now teaches for Indiana University — Purdue University / Indianapolis as a Senior Lecturer in Computer Science. He teaches Web programming, game development, and Freshman Computer Science classes.

Dedication

I dedicate this book to Jesus Christ, my personal savior, and to Heather, the joy in my life. I also dedicate this project to Benjamin, Jacob, Matthew, and Elizabeth. I love each of you.

Author's Acknowledgments

People often think of writing as a solo sport, but I know better. Thanks to Heather, for being amazing (again and again). Thank you Katie Feltman, for another interesting project, and for being a consistent friend. Thanks to Blair Pottenger for all your support on this book. You wrestled this monster into decent shape. Thanks very much to Heidi Unger for your editing help. It took more than a minute to win this one. Thank you Ronald Norman for the technical edit. You found a number of goofy errors that would have confused students. Thank you so much for your vigilance.

Thanks also to the many people at Wiley who the author never gets to meet. I appreciate your contributions. Thank you also to the open-source community which creates so many excellent tools. A big thanks to the IUPUI family for supporting me through this and so many other projects, especially Michele and Lingma.

Finally, thank you to my extended family — the Friday morning guys, and the Sunday evening families. I'm lucky to have a job where I get to publicly thank you for all you add to my life.

Publisher's Acknowledgments

We're proud of this book; please send us your comments at <http://dummies.custhelp.com>. For other comments, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

Some of the people who helped bring this book to market include the following:

Acquisitions, Editorial, and Media Development

Project Editor: Blair J. Pottenger

Acquisitions Editor: Katie Feltman

Copy Editor: Heidi Unger

Technical Editor: Ronald Norman

Editorial Manager: Kevin Kirschner

Media Development Project Manager:
Laura Moss-Hollister

Media Development Assistant Project Manager:
Jenny Swisher

Media Development Associate Producers:
Josh Frank, Marilyn Hummel, Douglas Kuhn,
and Shawn Patrick

Editorial Assistant: Amanda Graham

Sr. Editorial Assistant: Cherie Case

Composition Services

Project Coordinator: Katie Crocker

Layout and Graphics: Erin Zeltner

Proofreaders: John Greenough, Sossity R. Smith

Indexer: Potomac Indexing LLC

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Acquisitions Director

Mary C. Corder, Editorial Director

Publishing for Consumer Dummies

Diane Graves Steele, Vice President and Publisher

Composition Services

Debbie Stailey, Director of Composition Services

A Note About the Term HTML5

As this book was nearing completion, the World Wide Web Consortium (W3C) announced that the change of the upcoming version of HTML would not be HTML5, but simply HTML. They reasoned that a collaborative project like an HTML standard is an evolution rather than a strict milestone. When HTML5 is reasonably universal, there will be no need to call it HTML5, but simple HTML will do.

For the purposes of this book, it is important to distinguish between the current state of the art and the evolving standard that is the focus of this work. For that reason, I refer to the emerging standards as HTML5 to distinguish them from the older approaches to Web development, and keep the term HTML5 in the title.

Companion Resources On the Web

Be sure to check out my Web site for working examples of every code fragment in the book: www.aharrisbooks.net/h5qr.

Also check out this book's companion Web site at www.dummies.com/go/html5fdqr to access "Bonus Part 1: Using JavaScript" for a review (or preview) of computer programming in JavaScript. Programming is a complex business, and learning how to program in JavaScript deserves its own book; see my *JavaScript and AJAX For Dummies* (Wiley) or *HTML, XHTML, and CSS All-In-One For Dummies, 2nd edition* (Wiley) books for a more complete treatment.

Table of Contents

Part 1: Moving on to HTML5	1
A Quick History of HTML	2
A bit of ancient history	2
And the first browser war begins	2
A new challenger arises from the ashes	3
HTML 4 was getting old	4
Getting to Know the Real HTML5	4
HTML5 Is More than HTML!	5
HTML	6
CSS	6
JavaScript	7
Server technologies	7
Looking At Browser Features	8
Assessing your browser's capabilities	8
Checking for features in your code	9
Picking a Suitable Browser	12
Using Chrome Frame to Add Support to IE	13
 Part 2: HTML Foundations	 15
Exploring HTML and XHTML	16
Appreciating HTML	16
Emergence of XHTML	17
It's alive, and it's HTML5!	18
Setting up a basic HTML page	19
Fleshing Out Your Page	22
Adding images	22
Including links	24
Making lists and tables	26
Utilizing tables	28
Making a Form	30
Form structure tags	33
Constructing text input	34
Creating password fields	35
Erecting a multiline text box	35
Forming drop-down lists	36
Making checkboxes	37
Popping in radio buttons	38
Putting in action buttons	39
Validating Your Pages	40

Part 3: New or Changed HTML5 Elements 41

Semantic Page Elements.....	42
address.....	42
article.....	42
aside.....	43
footer.....	43
header.....	44
hgroup.....	44
menu.....	44
nav.....	45
section.....	45
Inline Semantic Elements.....	46
command.....	46
details.....	47
dfn.....	47
figcaption.....	47
figure.....	48
summary/details.....	48
time.....	48
wbr.....	49
Media Elements.....	49
audio.....	49
canvas.....	51
embed.....	53
source.....	54
svg.....	54
video.....	55
Ruby Elements.....	56
ruby.....	56
rt.....	56
rp.....	56

Part 4: New and Modified Form Elements..... 57

New Form Elements.....	58
datalist.....	58
fieldset.....	58
keygen.....	59
label.....	59
meter.....	60
output.....	61
progress.....	61
New Form Element Attributes.....	62
autofocus.....	62
pattern.....	62
placeholder.....	63
required.....	63
Validation.....	63

New Form Input Types	64
color	65
date	65
datetime	65
datetime-local	66
email	67
month	67
number	67
range	68
search	69
tel	69
time	70
url	70
week	70
Part 5: Formatting with CSS	71
A Quick Overview of CSS	72
Employing local styles	74
Making use of ids and classes	77
Managing levels of CSS	80
Managing the Appearance of Your Page	82
Comprehending hex colors	82
Editing text	85
Joining the border patrol	89
Putting in background images	90
Using Float Positioning	93
Getting to know the display types	95
Having a block party	96
Floating to a two-column look	97
Cleaning up the form	98
Using absolute positioning	100
Part 6: New and Improved CSS Elements	103
CSS3's New Selection Tools	104
Attribute selection	104
not	104
nth-child	104
Other new pseudo-classes	106
Downloadable Fonts and Text Support	106
@font-face	106
Column support	107
text-stroke	108
text-shadow	109
Flexible Box Layout Model	110
Creating a flexible box layout	110
Viewing a flexible box layout	111
... And now for a little reality	113

New Visual Elements	114
Color values	114
Gradients	115
Image borders	118
Reflections	119
Rounded corners	122
Shadows	122
Transformations	124
Transition animation	126
Transparency	128
Part 7: Changes in JavaScript.....	129
Behold: The New JavaScript Selection Options	130
document.getElementsByClassName()	130
document.getElementsByTagName()	130
document.querySelector()	131
document.querySelectorAll()	131
Data Options	131
Achieving offline cache	132
Local storage	134
WebSQL database	139
Miscellaneous New JavaScript Features	143
Geolocation	143
Notifications	146
Web sockets	148
Web workers	156
Part 8: Working with the Canvas	163
Canvas Basics	164
Setting up the canvas	164
Understanding how canvas works	165
Controlling Fill and Stroke Styles	166
Colors	166
Gradients	166
Patterns	169
Drawing Essential Shapes	171
Drawing rectangles	171
Drawing text	172
Enhancing shapes with shadows	173
Drawing More Complex Shapes	175
Line-drawing options	176
Making arcs and circles	179
Making quadratic curves	181
Producing a bezier curve	183

Images.....	184
Drawing an image on the canvas	185
Drawing part of an image	186
Manipulating Images with Transformations.....	187
Building a transformed image	188
Some key points about transformations.....	191
Using Animation	191
Basic structure of the animation loop.....	191
Creating the constants	192
Deploying the animation.....	193
Giving animation to the current frame.....	193
Moving an element.....	195
Now we're bouncing off the walls.....	197
Working with Pixel Manipulation.....	197

<i>Index.....</i>	<i>201</i>
--------------------------	-------------------

Part 1

Moving on to HTML5

HTML5 is the newest incarnation of the HTML family of languages. HTML, which stands for HyperText Markup Language, is one of the main reasons the Web is as powerful and useful as it is. HTML is a reasonably simple system of plain-text codes that provide the structure of all Web pages on the Internet.

In this part, you take a quick look at how HTML5 fits in the history of the Web, and put together a few tools you'll need to get started.



Be sure to check out my Web site for working examples of every code fragment in the book: www.aharrisbooks.net/h5qr.

In this part . . .

- ✓ **Looking at the History of HTML**
- ✓ **Understanding What HTML5 Is**
- ✓ **Running Tests for Browser Features**
- ✓ **Deciding on a Suitable Browser**
- ✓ **Utilizing Chrome Frame to Add Support to IE**

A Quick History of HTML

HTML is a key part of the Internet. It has a short but extremely vibrant history. In order to understand what HTML5 is about, it's useful to look at where it came from. The Internet (and the Web in particular) has been changing at a dizzying pace. HTML has been trying to keep up.

When HTML was first devised, it comprised a handful of tags, and HTML did little more than determine how a page was laid out. As the Web matured, many features were added. Today's Internet is still about documents, but it's also about applications. Today's Web sites are dynamic interactive applications.

The kinds of devices used on the Internet are changing, too. In the early days, only desktop computers used the Web. Now cellphones and mobile devices are among the most important players on the Web. They require a different way of thinking than the standard desk-based behemoths of a few years ago.

It's time for a fresh new set of standards that will help support the way people are using the Internet today. HTML5 is that set of standards.

A bit of ancient history

In the distant mists of time (1989) Tim Berners-Lee created a new system of connecting electronic documents. He devised a simple language that allowed document authors to link various documents together with limited formatting options. This language was called HTML.

At that point, the Internet existed, but it was mainly accessed by basic command-line programs, and was not easy to use. HTML (and some other underlying technologies) was designed from the beginning to be easy to work with, and to create documents that were easy for users to manage. The design of HTML was deliberately kept simple, so as many people as possible could participate in the process of building documents in this new format.

Of course, the Web took off in a very major way, and soon Web pages became ubiquitous. It became clear that the simple features in basic HTML were not enough to satisfy the interests of the many people who were now building Web pages.

And the first browser war begins . . .

As various organizations started building *Web browsers* (the tools that read HTML and display it to the user), they began competing by adding new HTML features. By 1993, the Mosaic browser included the ability to add images (which were not part of the original specification). Many browsers were being created by small teams all around the world, and each had its own set of new features.

By 1994, one platform emerged as the dominant browser. Netscape Navigator was a profoundly successful browser. At the same time, there were working groups

forming to address the lack of standards in the Web browser world. The most important of these groups was called the World Wide Web Consortium (W3C) headed by Tim Berners-Lee (the same guy who started all this mess). However, Netscape had such a dominant position that Netscape representatives often skipped the standards meetings and created whatever features they wanted.

Microsoft did not come into the browser world until 1995. Internet Explorer (IE) was designed to compete directly with Netscape's browser. For a time (sometimes called the first browser wars), Netscape and Microsoft were in an arms race, each trying to produce exclusive features that would steer developers toward their own vision of the Web.

While there was a standards body in place, the reality was both Netscape and Microsoft added whatever features they wanted and basically ignored the W3C. There was some small progress made on Web standards. HTML 2 was adopted as a standard in 1994/1995 (although none of the manufacturers stuck with it completely). HTML 3.2 was released in 1997, followed by HTML 4 in Spring of 1998.

By about the same time HTML 4 was gaining traction, it became clear that Microsoft was dominating the browser space. By 2002, Internet Explorer was used by approximately 95 percent of Internet users. With that kind of clout, the future of HTML was almost entirely in Microsoft's hands, and efforts of standards bodies were largely irrelevant. By any measure, Microsoft won the first browser war. Internet Explorer 6 (which used mainly HTML 4) was the only browser that really mattered, and there was very little innovation for several years.

A new challenger arises from the ashes

However, there were some new browsers that challenged Microsoft's dominance. The Firefox browser (first released in 2004) in particular was especially important, as it introduced a number of innovative features and followed most of the standards of the W3C working group. Firefox (and to a lesser extent other browsers like Apple's Safari, Opera, and eventually Google Chrome) shook up the Web. These other browsers tended to be more committed to following standards than IE was, and they prompted new versions of IE following a long era of stagnation. Even Microsoft began to at least pay lip service to the notion of standards, promising more standards compliance in each of the new versions of IE introduced. Some consider this the opening of the second browser war, with various developers competing for share of the browser market.

However, there is a difference this time around. The Web is no longer a novelty, but now a key part of business and society. A Web-based document is now held to the same visual standards as printed documents, and HTML 4 is simply not capable of easily meeting this standard. In fact, the entire notion of the Web as a series of documents is being challenged. Web *pages* are being replaced by Web *applications*. Much of what people now do on the Internet isn't about reading documents any more. Today, developers are using the Web itself as a programming interface.

HTML 4 was getting old

Changes in the Web required a change in the thinking about document standards. HTML 4 was clearly not up to the task of supporting modern Web development. The various proprietary tags added through the years added some visual flexibility, but not nearly enough. There was not a satisfying solution for page layout or font management. There was a set of features for entering form data, but these tools were limited and ugly. Most browsers featured a form of the JavaScript programming language, but the implementations varied wildly, and making a real application using Web technologies was a chancy proposition.

The W3C introduced XHTML in 2002 to address some of these concerns. XHTML was proposed as a version of HTML adhering to the stricter standards of the XML markup language. XHTML is much less forgiving than HTML, so if a page meets the stringent requirements of the standard, it is (presumably) well-behaved and predictable. Unfortunately, the idealism of the XHTML movement was never realized. Creating valid XHTML documents proved difficult enough that very few developers tried. Browsers rendered inaccurate XHTML code decently (if not perfectly). In fact, most browsers didn't really render XHTML at all, but quietly converted it to a form of HTML. There was little incentive for developers to adhere to XHTML standards (unless they were taking my class).

In order to get the functionality that was missing from HTML, many developers turned to plug-in technology like Java Applets and embedded Flash. Java never caught on as a client-side environment (although it remains extremely important in other applications) but Flash was very popular for a time. Unfortunately, Flash introduces problems of its own. The content of a Flash applet can only be modified by a Flash editor, and it cannot be read (at least easily) by search engines. Many of the new features of HTML5 (particularly the font support and the canvas tag) can be seen as a direct response to Flash.

The W3C moved to create a new form of XHTML called XHTML 2.0, but in the mean time, a second group called WHATWG (Web Hypertext Application Technology Working Group) began working on their own competing standard, which came to be known as HTML5. The main reason for these competing standards was a sense that XHTML was too rigid, and was still focused on HTML as a document language. Part of the motivation for HTML5 was to create a framework for building Web applications that would really be used by developers. Eventually, W3C dropped support for XHTML 2 and is now supporting the WHATWG proposal, so HTML5 appears to be the next standard.

Getting to Know the Real HTML5

The WHATWG group seems to have learned a few lessons from history. The design of HTML5 indicates these priorities: