

Wikipedia

THE MISSING MANUAL[®]



The book
that should
have been
in the box[®]

"The Missing Manual
series is simply the
most intelligent and
usable series of
guidebooks..."

—Kevin Kelly,
co-founder of Wired

POGUE PRESS[™]
O'REILLY[®]

John Broughton

Wikipedia: The Missing Manual

by John Broughton

Copyright © 2008 O'Reilly Media. All rights reserved.

Printed in the United States of America.

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

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

Editors: Nan Barber and Peter Meyers

Copy Editor: Sohaila Abdulali and Jill Steinberg

Production Editor: Nellie McKesson

Proofreader: Nellie McKesson

Indexer: Dawn Frausto

Cover Designer: David Freedman

Illustrators: Robert Romano and Jessamyn Read

Printing History:

January 2008: First Edition

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Wikipedia: The Missing Manual*, The Missing Manual logo, Pogue Press, and the Pogue Press logo and related trade dress are trademarks of O'Reilly Media, Inc.

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

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



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN-10: 0-596-51516-2

ISBN-13: 978-0-596-51516-4

[M]

This excerpt is protected by copyright law. It is your responsibility to obtain permissions necessary for any proposed use of this material. Please direct your inquiries to permissions@oreilly.com.

Adding and Deleting Scripts

To use user scripts, you create a personal JavaScript page, which only you and Wikipedia administrators can edit. On this page, you put the user scripts that you want to use, or pointers to other pages where JavaScript code is stored (which Wikipedia uses to add code to your page).

When you want to stop using a script, you edit that personal page, deleting the JavaScript code or pointer, or marking it as non-executable information. That's called *commenting it out*, and it just means you're telling the Wikipedia software not to follow those instructions.

NOTE

In late 2007, Wikipedia added a new feature that lets you implement user scripts with just a couple of clicks. On the My Preferences page, check the Gadgets tab to see if the script you want to add is listed on that tab. If so, you can save time by turning it on that tab, rather than following the instructions in this section. (For details, see page 399.)

Your Personal JavaScript Page

Creating your JavaScript page is as easy as creating a subpage in your own user space, as described in Chapter 3. See page 56 if you need a refresher on subpages. This chapter assumes that you're using the standard Monobook skin for your Wikipedia window (page 388), and so you'll name your personal page *monobook.js*. (If you use a different skin, see the box on page 408.)

When you're ready to do the JavaScript jive, go to your user page, and follow these steps:

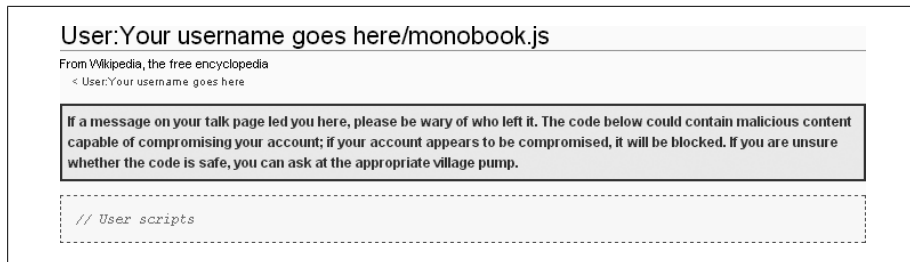


Figure 21-1. All monobook.js pages have the same warning at the top – don’t copy a script just because someone else leaves a note on your user talk page about what a neat-cool tool they’ve created. (Leaving uninvited, non-personal postings on user talk pages is considered spam, and fortunately it doesn’t happen very often.) If you use JavaScript code someone else wrote, make sure it’s someone you trust.

1. On your user page, click “edit this page”. In the edit box, type either `*[[User:Your-UsernameGoesHere/monobook.js]]` or simply `*[[/monobook.js]]`. Then add an edit summary like *Creating link to new monobook.js page*, do a preview, and then save your edit.

You’ve just added a link to your soon-to-be-created JavaScript page. The link to the new page is red, since it doesn’t yet exist.

2. Click the red link to the *monobook.js* page, and, at the top of the edit box, add the following: `// User scripts`. (In JavaScript code, anything that begins with double slashes is a comment. You’ll see the “User scripts” comment, as shown in Figure 21-1.) Add an edit summary (like *Creating initial page*), preview the page (to reinforce the habit), and then save the page.

By saving the page with only a comment, initially, you’re creating a version of the page that you can revert to if something goes wrong. That’s extremely unlikely, but it doesn’t hurt to have the option.

Adding a Script

As discussed in Chapter 7 (page 124), when you’re evaluating a user account to determine whether it’s a vandalism-only account, it’s helpful to look at more than the user talk page of the editor. You’ll often want to look at individual edits (the *User contributions* page) and the block log, for instance. This tutorial will show you how to implement some JavaScript that add tabs, making that research easier.

1. Go to the page *Wikipedia:WikiProject User scripts/Scripts/User tabs* (Figure 21-2), and copy all the text within the dotted box.

This page has no “edit this page” tab, meaning it’s protected. Don’t copy JavaScript or any other code from an unprotected page, since someone may have tampered with it.



Figure 21-2. The script at the page Wikipedia:WikiProject User scripts/Scripts/User tabs has three full lines of comment (two at the top, one at the bottom); these have two slashes at the beginning. The actual code that will be executed is the middle fifteen lines. The blank spaces at the beginning of any line are optional—they're there to make the script easier for humans to read. Computers don't care one way or another. (The blank spaces at the beginnings of lines are why the text goes off-screen on the right; that's normal for Wikipedia wikitext.)

2. Open your own *monobook.js* page, and then, in the edit box, paste the text that you copied in step 1.
If you haven't created a *monobook.js* page yet, see page 402.
3. Change `[[User:Where/usertabs]]` to `[[Wikipedia:WikiProject User scripts/Scripts/User tabs]]`, to show where you got the script.
Your edit box should look like Figure 21-3.
4. Add an edit summary (Adding JavaScript for user tabs, for example) and then click the "Show preview" button.
The preview should look like Figure 21-4. If everything looks as you expect (as is the case here, there's no visible change), save the page. If something looks wrong, recheck the code that you added to the page.
5. After you save the change, you should bypass your cache so that Wikipedia isn't using an old version of your *monobook.js* (see page 409). Then go to a page where you expect the change to appear—for example, click on the "my talk" link on the upper right, and check if the change has happened.

Editing User:Your username goes here/monobook.js

From Wikipedia, the free encyclopedia

Tip: Use the 'Show preview' button to test your new css/js before saving. Have a cool style? Share it in the Gallery of user

```
// User scripts
// Please leave the following line
// [[Wikipedia:WikiProject User scripts/Scripts/User tabs]]

addOnloadHook(function() {
    if (wgTitle.indexOf("/") != -1 || document.title.indexOf("- History -") != -1)
        return;
    if (wgCanonicalNamespace == "User" || wgCanonicalNamespace == "User_talk") {
        var username = encodeURIComponent( wgTitle );
        addPortletLink("p-cactions", wgServer + "/wiki/Special:Contributions/" + username,
            "contributions");
        addPortletLink("p-cactions", "http://tools.wikimedia.de/~interiot/cgi-bin/Toolbox.php?site=en.wikipedia.org", "Edit count", "ca-editcount", "Edit count from Interiot");
        addPortletLink("p-cactions", wgServer + "/w/index.php?title=Special:Log&type=" +
            "ca-pagemoves", "Page moves by this user");
        addPortletLink("p-cactions", wgServer + "/w/index.php?title=Special:Log&type=" +
            "received", "ca-blog", "Blocks received by this user");
        addPortletLink("p-cactions", wgServer + "/w/index.php?title=Special:Prefixindex&namespace=2", "Userspace", "", "List of pages in this user's userspace");
        addPortletLink("p-cactions", wgServer + "/w/index.php?title=Special:Log&type=" +
            "ca-pagemoves", "Page moves by this user");
    }
});
```

Figure 21-3. In editing mode, monobook.js pages look different from regular pages. There's no edit toolbar, because there's no need for buttons for inserting a signature, or a table, or other text. There's also a message about using the "Show preview" button to test the added JavaScript before saving the page.

As shown in Figure 21-5, it indeed has.

If you still can't see what you expect after bypassing your cache, then ask for assistance at the Help desk (shortcut: WP:HD).

Preventing Code Overload

It's good practice to put anything other than a very short script into a separate page, with a name ending in .js, so that only you—and administrators—can edit it. That way, you don't end up with a very long, difficult-to-understand and difficult-to-edit *monobook.js* page.

Suppose, for example, you put the "User tabs" JavaScript from the previous steps, into a new page called *User:Your username goes here/UserTabs.js*. That keeps it out of your *monobook.js* page, but it also means that Wikipedia doesn't recognize it as JavaScript you actually want to use. In fact, when you go into edit mode at a new page, to paste in the code, you see a warning message (Figure 21-7) that tells you that Wikipedia can't match the file name against any of the skins you could have chosen (see page 388), and therefore won't automatically execute the script when you're logged on.

To actually use the JavaScript at your new *User:Your username goes here/UserTabs.js* page, you have to tell Wikipedia to *import* into your *monobook.js* page. When the Wikipedia software reads your *monobook.js* page, it reads the import script, and learns

Editing User:Your username goes here/monobook.js

From Wikipedia, the free encyclopedia

You are testing your user JavaScript right on this page. Remember it has not yet been saved!

Preview

Remember that this is only a preview; any changes have not yet been saved!

```
// User scripts
// Please leave the following line
// [[Wikipedia:WikiProject User scripts/Scripts/User tabs]]

addOnloadHook(function() {
    if (wgTitle.indexOf("/") != -1 || document.title.indexOf("- History -") != -1) //n
        return;
    if (wgCanonicalNamespace == "User" || wgCanonicalNamespace == "User_talk") {
        var username = encodeURIComponent( wgTitle );
        addPortletLink("p-cactions", wgServer + "/wiki/Special:Contributions/" + username
        contributions");
        addPortletLink("p-cactions", "http://tools.wikimedia.de/~interiot/cgi-bin/Tool1/
+ "&site=en.wikipedia.org", "Edit count", "ca-editcount", "Edit count from Interiot's
        addPortletLink("p-cactions", wgServer + "/w/index.php?title=Special:Log&type=mov
        "ca-pagemoves", "Page moves by this user");
        addPortletLink("p-cactions", wgServer + "/w/index.php?title=Special:Log&type=blo
        received", "ca-blog", "Blocks received by this user");
```

Figure 21-4. The nice thing about previewing a change to your personal JavaScript page is that the JavaScript is implemented immediately. If it should change the monobook.js page, you see that. In this case, the new tabs will only show on a user page, so the preview doesn't tell you how you're doing.

where to go to get the actual JavaScript. Figure 21-7 shows a bunch of import scripts on an editor's monobook.js page.

NOTE

In Figure 21-7, the line second from the bottom has two slash marks in front, making the line into a comment. In geek speak, the line is *commented out*, so the script won't run. To reverse that in the future, making it functional again, just delete the slashes.

Importing Multi-Function Scripts of Other Editors

Several Wikipedia editors have created big, fabulously useful, multi-function scripts. You absolutely don't want to cut and paste these scripts, even to a separate page of your own, because you need the editor who created it to maintain it. You want to get enhancements to the code as they occur. By using the import function, you get the benefit of those updates (as do the hundreds of other editors using the same script).

WARNING

As always, don't trust any page that isn't protected from editing by regular editors. Don't ever import from an unprotected page.

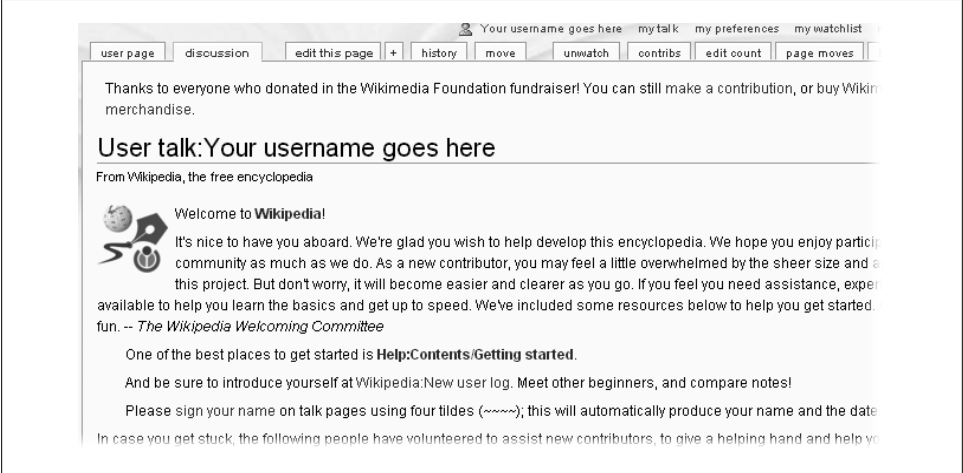


Figure 21-5. Now a number of additional tabs appear whenever you go to a user or user talk page. If you decide you don't want one or more of these, all you need to do is comment out the corresponding line on your `monobook.js` page (put two slashes at the very beginning), and that tab won't appear any more.



Figure 21-6. When you create a page whose name ends with `.js` or `.css`, Wikipedia checks to see if that's the name of a skin (see page 388). If not, it gives you this warning. In this case, you can ignore the warning. But if you typed (for example) `MonoBook.js` rather than `monobook.js`, earlier in the chapter, the misspelling would be critical and the warning would be relevant.

Removing Scripts

As you can see from reading this chapter, editing your `monobook.js` page is no big deal. It's no harder than cutting and pasting text to any Wikipedia article. You can remove a function by deleting it, or by commenting it out (putting a pair of slashes at the beginning of each line). Commenting out works best for cases where you're importing a function (see Figure 21-7) and there's just one line to comment out. If you have 10 or 15 lines of code that you copied to your `monobook.js` page that you don't want any more, the page will be a lot neater if you delete it all, perhaps leaving a single comment line about where you found the code, in case you want to copy it back in at some later point.

```
importScript('User:Ais523/votesymbols.js'); //[[User:Ais523/votesymbols.js]]
importScript('User:Ais523/topcontrib.js'); //[[User:Ais523/topcontrib.js]]
importScript('User:Ais523/formatedit.js'); //[[User:Ais523/formatedit.js]]
importScript('User:Ais523/contribcalendar.js'); //[[User:Ais523/contribcalendar.js]]
importScript('User:Ais523/highlightmyname2.js'); //[[User:Ais523/highlightmyname2.js]]
importScript('User:Ais523/catwatch.js'); //[[User:Ais523/catwatch.js]]
importScript('User:Ais523/sandbox.js'); //[[User:Ais523/sandbox.js]]
importScript('User:Ais523/watchlistnotifier.js'); //[[User:Ais523/watchlistnotifier.js]]
importScript('User:Ais523/adminrights.js'); //[[User:Ais523/adminrights.js]]
//importScript('User:Ais523/watchlistei.js'); //[[User:Ais523/watchlistei.js]]
importScript('User:Ais523/editsection0tab.js'); //[[User:Ais523/editsection0tab.js]]
```

Figure 21-7. The page `User:Ais523/monobook.js` has a number of JavaScript functions (user scripts) added via the `importScript` command, rather than pasting all the JavaScript into the page. One advantage of `importScript` is that you get a neat list of all your user scripts, rather than a mass of code. Separating functions by putting them on separate pages also makes it easier to share scripts with other editors.

POWER USERS' CLINIC

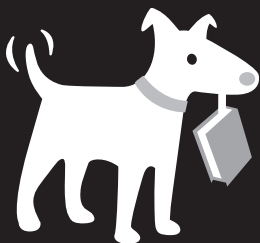
If You Don't Use the Monobook Skin

As discussed in the previous chapter (page 388), in your My Preferences page, on the Skin tab, you have an option of selecting one of seven skins, which rearrange the elements and formatting on each Wikipedia page that you view. The tab has buttons to click to see a preview of each skin, so you can get a clear sense of what a change of skins can do.

You have no guarantee that a user script that works in Monobook—which is what almost all editors use, and the norm for testing user scripts—will also work if you use another skin. But if you want to try using a different skin, understand that each skin looks for a specific, and different, personal file when it implements your personal JavaScript. Just create that page for your user scripts instead of or in addition to *monobook.js*.

The files used by the other six skins are:

- Chick: *chick.js*
- Classic skin: *standard.js*
- Cologne Blue: *cologneblue.js*
- MySkin: *myskin.js*
- Nostalgia: *nostalgia.js*
- Simple: *simple.js*



THE MISSING MANUAL[®]

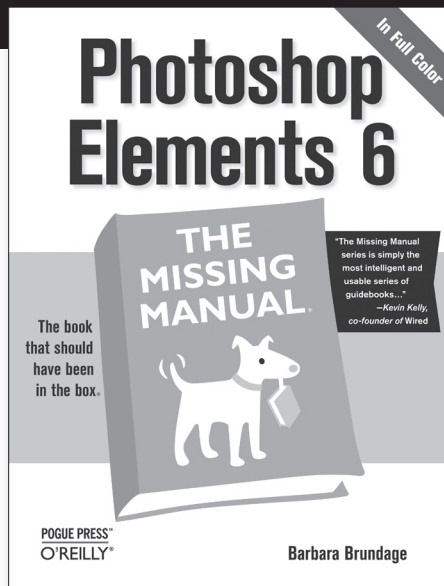
The book that should have been in the box[®]

Answers found here!

Missing Manuals deliver complete and comprehensive coverage of the most popular operating systems, applications and multimedia devices. Written in a unique, witty style that helps you learn quickly, these entertaining books explain every feature clearly and thoroughly, helping you master every aspect of your software.

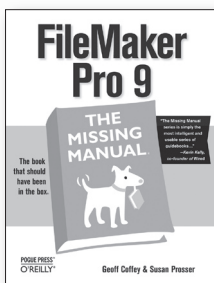
Photoshop Elements 6: The Missing Manual

By Barbara Brundage, ISBN 9780596514440
\$39.99 US, 47.99 CAN



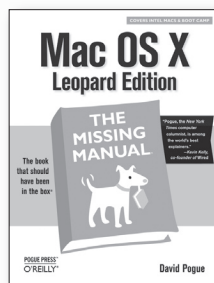
Flash CS3: The Missing Manual

By Chris Grover & E.A. Moore
ISBN 9780596510442
\$34.99 US, 45.99 CAN



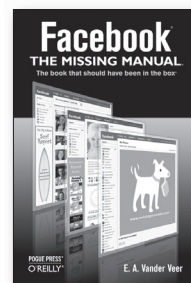
Filemaker Pro 9: The Missing Manual

By Geoff Coffey & Susan Prosser
ISBN 9780596514136
\$34.99 US, 41.99 CAN



Mac OS X: The Missing Manual, Leopard Edition

By David Pogue
ISBN 9780596529529
\$34.99 US, 34.99 CAN



Facebook: The Missing Manual

By E.A. Vander Veer
ISBN 9780596517694
\$19.99 US, 19.99 CAN

Look for these and other Missing Manuals
at your favorite bookstore or online.

www.missingmanuals.com

POGUE PRESS[™]
O'REILLY[®]