

Extracted from:

HTML5 and CSS3

Develop with Tomorrow's Standards Today

This PDF file contains pages extracted from HTML5 and CSS3, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

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, or otherwise, without the prior consent of the publisher.

The
Pragmatic
Programmers

HTML5 & CSS3

Develop with Tomorrow's
Standards Today



Brian P. Hogan

Edited by Susannah Davidson Pfalzer

11

Providing Navigation Hints with ARIA Roles

Most websites share a common structure: there's a header, a navigation section, some main content, and a footer. Most of these sites are coded just like that, in a linear fashion. Unfortunately, this means that a screen reader may have to read the site to its user in that order. Since most sites repeat the same header and navigation on each page, the user will have to hear these elements each time they visit another page.

The recommended fix is to provide a hidden “skip navigation” link that screen readers will read aloud, which simply links to an anchor somewhere near the main content. However, that's not something that's built in, and it's not something that everyone knows how (or remembers) to do.

HTML5's new `role` attribute lets us assign a “responsibility” to each element on your page. A screen reader can then very easily parse the page and categorize all of those responsibilities so that you can create a simple index for the page. For example, it can find all the navigation roles on the page and present them to the user so they can quickly navigate around your application.

These roles come from the WIA-ARIA specification³ and have been incorporated into the HTML5 specification. There are two specific classifications of roles that you can put to use right now: landmark roles and document roles.

Landmark Roles

Landmark roles identify “points of interest” on your site, such as the banner, search area, or navigation that screen readers can quickly identify.

3. <http://www.w3.org/WAI/PF/aria/roles>

Role	Use
banner	Identifies the banner area of your page
search	Identifies the search area of your page
navigation	Identifies navigational elements on your page
main	Identifies where your page's main content begins
contentinfo	Identifies where information about the content exists, such as copyright information and publication date
complementary	Identifies content on a page that complements the main content but is meaningful on its own
application	Identifies a region of a page that contains a web application as opposed to a web document

We can apply a few of these roles to the AwesomeCo blog template we worked on in *Redefining a Blog Using Semantic Markup*, on page 29.

For the overall header, let's apply the banner role like this:

[Download](#) html5_aria/blog/index.html

```
<header id="page_header" role="banner">
  <h1>AwesomeCo Blog!</h1>
</header>
```

All that's needed is the addition of the `role="banner"` to the existing header tag.

We can identify our navigation the same way:

[Download](#) html5_aria/blog/index.html

```
<nav role="navigation">
  <ul>
    <li><a href="/">Latest Posts</a></li>
    <li><a href="/archives">Archives</a></li>
    <li><a href="/contributors">Contributors</a></li>
    <li><a href="/contact">Contact Us</a></li>
  </ul>
</nav>
```

The HTML5 specification says that some elements have default roles and can't be overridden. The `nav` element must have the role of navigation and technically doesn't need to be specified. Screen readers aren't quite ready to accept that default yet, but many of them do understand these ARIA roles.

Our main and sidebar regions can be identified as follows:

[Download](#) `html5_aria/blog/index.html`

```
<section id="posts" role="main">
</section>
```

[Download](#) `html5_aria/blog/index.html`

```
<section id="sidebar" role="complementary">

  <nav>
    <h3>Archives</h3>
    <ul>
      <li><a href="2010/10">October 2010</a></li>
      <li><a href="2010/09">September 2010</a></li>
      <li><a href="2010/08">August 2010</a></li>
      <li><a href="2010/07">July 2010</a></li>
      <li><a href="2010/06">June 2010</a></li>
      <li><a href="2010/05">May 2010</a></li>
      <li><a href="2010/04">April 2010</a></li>
      <li><a href="2010/03">March 2010</a></li>
      <li><a href="2010/02">February 2010</a></li>
      <li><a href="2010/01">January 2010</a></li>
    </ul>
  </nav>
</section> <!-- sidebar -->
```

We identify the publication and copyright info in our footer using the `contentinfo` role like this:

[Download](#) `html5_aria/blog/index.html`

```
<footer id="page_footer" role="contentinfo">
  <p>&copy; 2010 AwesomeCo.</p>
</footer> <!-- footer -->
```

If we had a search for our blog, we could identify that region as well. Now that we've identified the landmarks, let's take this a step further and help identify some of the document elements.

Document Structure Roles

Document structure roles help screen readers identify parts of static content easily, which can help better organize content for navigation.

Role	Use
document	Identifies a region containing document content, as opposed to application content.
article	Identifies a composition that forms an independent part of a document.
definition	Identifies a definition of a term or subject.
directory	Identifies a list of references to a group, like a table of contents. Used for static content.
heading	Identifies a heading for a section of a page.
img	Identifies a section that contains elements of an image. This may be image elements as well as captions and descriptive text.
list	Identifies a group of noninteractive list items.
listitem	Identifies a single member of a group of noninteractive list items.
math	Identifies a mathematical expression.
note	Identifies content that is parenthetical or ancillary to the main content of the resource.
presentation	Identifies content that is for presentation and can be ignored by assistive technology.
row	Identifies a row of cells in a grid.
rowheader	Identifies a cell containing header information for a row in a grid.

Many of the document roles are implicitly defined by HTML tags, such as articles and headers. However, the document role isn't, and it's a very helpful role, especially in applications with a mix of dynamic and static content. For example, a web-based email client may have the document role attached to the element that contains the body of the email message. This is useful because screen readers often have different methods for navigating using the keyboard. When the screen reader's focus is on an application element, it may need to allow keypresses through to the web application. However, when the focus is on static content, it could allow the screen reader's key bindings to work differently.

We can apply the document role to our blog by adding it to the body element.

[Download](#) `html5_aria/blog/index.html`

```
<body role="document">
```



Joe Asks...

Do We Need These Landmark Roles If We Have Elements Such As nav and header?

The landmark roles may at first seem redundant, but they provide you with the flexibility you need for situations where you can't use the new elements.

Using the search role, you can direct your users to the region of the page that not only contains the search field but also links to a site map, a drop-down list of "quick links," or other elements that will help your users find information quickly, as opposed to just directing them to the actual search field.

There are also a lot more roles introduced by the specification than there are new elements and form controls.

This can help ensure that a screen reader will treat this page as static content.

Falling Back

These roles are already usable on the latest browsers with the latest screen readers, so you can start working with them now. Browsers that don't support them are just going to ignore them, so you're really only helping those people who can use them.

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Home Page for HTML5 and CSS3

<http://pragprog.com/titles/bhh5>

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

<http://pragprog.com/updates>

Be notified when updates and new books become available.

Join the Community

<http://pragprog.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

<http://pragprog.com/news>

Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/bhh5.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)