

**CONTENTS INCLUDE:**

- HTML Basics
- HTML vs XHTML
- Validation
- Useful Open Source Tools
- Page Structure Elements
- Key Structural Elements and more...

# Core HTML

By Andy Harris

## HTML BASICS

HTML and XHTML are the foundation of all web development. HTML is used as the graphical user interface in client-side programs written in JavaScript. Server-side languages like PHP and Java also receive data from web pages and use HTML as the output mechanism. The emerging Ajax technologies likewise use HTML and XHTML as their visual engine. HTML was once a very loosely-defined language with very little standardization, but as it has become more important, the need for standards has become more apparent. Regardless of whether you choose to write HTML or XHTML, understanding the current standards will help you provide a solid foundation that will simplify all your other web coding. Fortunately HTML and XHTML are actually simpler than they used to be, because much of the functionality has moved to CSS.

### Common Elements

Every page (HTML or XHTML shares certain elements in common.) All are essentially plain text files, with the .html extension. HTML files should not be created with a word processor, but in some type of editor that creates plain text. Every page has a large container (HTML or XHTML) and two major subcontainers, the head and the body. The head area contains information useful behind the scenes, such as CSS formatting instructions and JavaScript code. The body contains the part of the page that is visible to the user.

### Tags and Attributes

An HTML document is based on the notion of tags. A tag is a piece of text inside angle brackets (<>). Tags typically have a beginning and an end, and usually contain some sort of text inside them. For example, a paragraph is normally denoted like this:

```
<p>
  This is my paragraph.
</p>
```

The <p> indicates the beginning of a paragraph. Text is then placed inside the tag, and the end of the paragraph is denoted by an end tag, which is similar to the start tag but with a slash (</p>.) It is common to indent content in a multi-line tag, but it is also legal to place tags on the same line:

```
<p>This is my paragraph.</p>
```

Tags are sometimes enhanced by attributes, which are name value pairs that modify the tag. For example, the <img> tag (used to embed an image into a page) usually includes the following attributes:

```
<img src = "myPic.jpg"
      Alt = "this is my picture" />
```

The src attribute describes where the image file can be found, and the alt attribute describes alternate text that is displayed if the image is unavailable.

### Nested tags

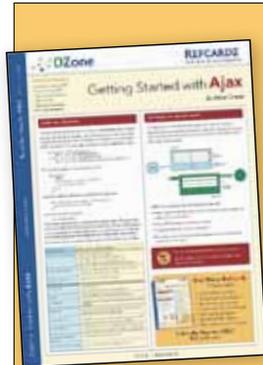
Tags can be (and frequently are) nested inside each other. Tags cannot overlap, so <a><b></a></b> is not legal, but <a><b></b></a> is fine.

## HTML VS XHTML

HTML has been around for some time. While it has done its job admirably, that job has expanded far more than anybody expected. Early HTML had very limited layout support. Browser manufacturers added many competing standards and web developers came up with clever workarounds, but the result is a lack of standards and frustration for web developers. The latest web standards (XHTML and the emerging HTML 5.0 standard) go back to the original purpose of HTML: to describe the structure of the data only, and leave all formatting to CSS (Please see the DZone CSS Refcard Series). **XHTML is nothing more than HTML code conforming to the stricter standards of XML.** The same style guidelines are appropriate whether you write in HTML or XHTML (but they tend to be enforced in XHTML):

- Use a doctype to describe the language (described below)
- Write all code in lowercase letters
- Encase all attribute values in double quotes
- Each tag must have an end specified. This is normally done with an ending tag, but a special case allows for non-content tags.

Most of the requirements of XHTML turn out to be good practice whether you write HTML or XHTML. I recommend



## Get More Refcardz (They're free!)

- Authoritative content
- Designed for developers
- Written by top experts
- Latest tools & technologies
- Hot tips & examples
- Bonus content online
- New issue every 1-2 weeks

**Subscribe Now for FREE!**  
[Refcardz.com](http://Refcardz.com)

using XHTML strict so you can validate your code and know it follows the strictest standards.

XHTML has a number of flavors. The strict type is recommended, as it is the most up-to-date standard which will produce the most predictable results. You can also use a transitional type (which allows deprecated HTML tags) and a frameset type, which allows you to add frames. For most applications, the strict type is preferred.

### HTML Template

The following code can be copied and pasted to form the foundation of a basic web page:

```
<html>
<head>
  <title></title>
</head>

<body>

</body>
</html>
```

### XHTML Template

The XHTML template is a bit more complex, so it's common to keep a copy on your desktop for quick copy – and paste work, or to define it as a starting template in your editor.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="EN" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/xml;
  charset=utf-8" />
  <title></title>
</head>

<body>

</body>
</html>
```

### VALIDATION

The structure of your web pages is critical to the success of programs based on those pages, so use a validating tool to ensure you haven't missed anything.

Validating Tool	Description
WC3	The most commonly used validator is online at <a href="http://validator.w3.org">http://validator.w3.org</a> This free tool checks your page against the doctype you specify and ensures you are following the standards. This acts as a 'spell-checker' for your code and warns you if you made an error like forgetting to close a tag.
HTML Tidy	There's an outstanding free tool called HTML tidy which not only checks your pages for validity, but also fixes most errors automatically. Download this tool at <a href="http://tidy.sourceforge.net/">http://tidy.sourceforge.net/</a> or (better) use the HTML validator extension to build tidy into your browser.
HTML Validator extension	The extension mechanism of Firefox makes it a critical tool for web developers. The HTML Validator extension is an invaluable tool. It automatically checks any page you view in your browser against both the w3 validation engine and tidy. It can instantly find errors, and repair them on the spot with tidy. With this free extension available at <a href="http://users.skynet.be/mgueury/mozilla/">http://users.skynet.be/mgueury/mozilla/</a> , there's no good reason not to validate your code.

### USEFUL OPEN SOURCE TOOLS

Some of the best tools for web development are available through the open source community at no cost at all. Consider these application as part of your HTML toolkit:

Open Source Tool	Description
Aptana	<a href="http://www.aptana.com/">http://www.aptana.com/</a> This free programmer's editor (based on Eclipse) is a full-blown IDE customized for HTML / XHTML, CSS, JavaScript, and Ajax. It offers code completion, syntax highlighting, and FTP support within the editor.

Web Developer Toolbar	<a href="https://www.addons.mozilla.org/en-US/firefox/addon/60">https://www.addons.mozilla.org/en-US/firefox/addon/60</a> This Firefox extension adds numerous debugging and web development tools to your browser.
Firebug	<a href="https://addons.mozilla.org/en-US/firefox/addon/1843">https://addons.mozilla.org/en-US/firefox/addon/1843</a> is an add-on that adds full debugging capabilities to the browser. The firebug lite version even works with IE.

### PAGE STRUCTURE ELEMENTS

The following elements are part of every web page.

Element	Description
<html></html>	Surrounds the entire page
<head></head>	Contains header information (metadata, CSS styles, JavaScript code)
<title></title>	Holds the page title normally displayed in the title bar and used in search results
<body></body>	Contains the main body text. All parts of the page normally visible are in the body

### KEY STRUCTURAL ELEMENTS

Most pages contain the following key structural elements:

Element	Name	Description
<h1></h1>	Heading 1	Reserved for strongest emphasis
<h2></h2>	Heading 2	Secondary level heading. Headings go down to level 6, but <h1> through <h3> are most common
<p></p>	Paragraph	Most of the body of a page should be enclosed in paragraphs
<div></div>	Division	Similar to a paragraph, but normally marks a section of a page. Divs usually contain paragraphs

### LISTS AND DATA

Web pages frequently incorporate structured data so HTML includes several useful list and table tags:

Element	Name	Description
<ul></ul>	Unordered list	Normally these lists feature bullets (but that can be changed with CSS)
<ol></ol>	Ordered list	These usually are numbered, but this can be changed with CSS
<li></li>	List item	Used to describe a list item in an unordered list or an ordered list
<dl></dl>	Definition list	Used for lists with name-value pairs
<dt></dt>	Definition term	The name in a name-value pair. Used in definition lists
<dd></dd>	Definition description	The value (or definition) of a name – value pair
<table></table>	Table	Defines beginning and end of a table
<tr></tr>	Table row	Defines a table row. A table normally consists of several <tr> pairs (one per row)
<td></td>	Table data	Indicates data in a table cell. <td> tags occur within <tr> (which occur within <table>)
<th></th>	Table heading	Indicates a table cell to be treated as a heading with special formatting

Visit <http://www.aharrisbooks.net/dzone/listTable.html> for an example. Use view source to see the XHTML code.

### Standard List Types

HTML supports three primary list types. Ordered lists and unordered lists are the primary list types. By default, ordered lists use numeric identifiers, and unordered lists use bullets.

However, you can use the list-style-type CSS attribute to change the list marker to one of several types.

```
<ol>
  <li>uno</li>
  <li>dos</li>
  <li>tres</li>
</ol>
```

Lists can be nested inside each other

```
<ul>
  <li>English
    <ol>
      <li>One</li>
      <li>Two</li>
      <li>Three</li>
    </ol>
  </li>
  <li>Spanish
    <ol>
      <li>uno</li>
      <li>dos</li>
      <li>tres</li>
    </ol>
  </li>
</ul>
```

### Definition lists

The special definition list is used for name / value pairs. The definition term (dt) is a word or phrase that is used as the list marker, and the definition data is normally a paragraph:

```
<h2>Types of list</h2>
<dl>
  <dt>Unordered list</dt>
  <dd>Normally used for bulleted lists, where the order of data is not important. </dd>

  <dt>Ordered lists</dt>
  <dd>Normally use numbered items, for example a list of instructions where the order is significant.</dd>

  <dt>Definition list</dt>
  <dd>Used to describe a term and definition. Often a good alternative to a two-column table</dd>
</dl>
```

### Use of tables

Tables were used in the past to overcome the page-layout shortcomings of HTML. That use is now deprecated in favor of CSS-based layout. Use tables only as they were intended – to display tabular data.

A table mainly consists of a series of table rows (tr.) Each table row consists of a number of table data (td) elements. The table heading (th) element can be used to indicate a table cell should be marked as a heading.

The rowspan and colspan attributes can be used to make a cell span more than one row or column.

Each row of a table should have the same number of columns, and each column should have the same number of rows. Use of the span attribute may require adjustment to other rows or columns.

```
<table border = "1">
  <tr>
    <th>&nbsp;</th>
    <th>English</th>
    <th>Spanish</th>
  </tr>

  <tr>
    <th>1</th>
    <td>One</td>
    <td>Uno</td>
  </tr>

  <tr>
    <th>2</th>
    <td>Two</td>
    <td>Dos</td>
  </tr>
</table>
```

## LINKS AND IMAGES

Links and images are both used to incorporate external resources into a page. Both are reliant on URIs (Universal Resource Indicators), commonly referred to as URLs or addresses.

### <a> (anchor)

The anchor tag is used to provide the basic web link:

```
<a href = "http://www.google.com">link to Google</a>
```

In this example, <http://www.google.com> is the site to be visited. The text "link to Google" will be highlighted as a link.

### absolute and relative references

Links can be absolute references containing an entire url including the http: protocol indicator. <http://www.aharrisbooks.net> goes directly to my site from any page on the internet.

A relative reference leaves out the http:// business. The browser assumes the same directory on the same server as the referring page. If this link: [<a href = "xfd">XHTML for Dummies</a>](#) is on my main site, it will take you to <http://www.aharrisbooks.net/xfd>.

### <link>

The link tag is used primarily to pull in external CSS files:

```
<link rel = "stylesheet"
      type = "text/css"
      href = "mySheet.css" />
```

### <img>

The img tag is used in to attach an image. Valid formats are .jpg, .png, and .gif. An image should always be accompanied by an alt attribute describing the contents of the image.

```
<img src = http://www.cs.iupui.edu/~aharris/face.gif
      alt = "me before shaving" />
```

Image formatting attributes (height, width, and align) are deprecated in favor of CSS.

## SPECIALTY MARKUP

HTML / XHTML includes several specialty tags. These are used to describe special purpose text. They have default styling, but of course the styles can be modified with CSS.

### <quote>

The quote tag is intended to display a single line quote:

```
<quote>Now is the time for all good men to come to the aid of their country</quote>
```

Quote is an inline tag. If you need a block level quote, use <blockquote>.

### <pre>

The <pre> tag is used for pre-formatted text. It is sometimes used for code listings or ASCII art because it preserves carriage returns. Pre-formatted text is usually displayed in a fixed-width font.

```
<pre>
for i in range(10):
    print i
</pre>
```

### <code>

The code format is used to manage pre-formatted text, especially code listings. It is very similar to pre.

```
<code>
while i < 10:
  i += 1
  print i
</code>
```

### <blockquote>

This tag is used to mark multi-line quotes. Frequently it is set off with special fonts and indentation through CSS. It is (not surprisingly) a block-level tag.

```
<blockquote>
Quoth the raven:
Nevermore
</blockquote>
```

### <span>

The span tag is a vanilla inline tag. It has no particular formatting of its own. It is intended to be used with a class or ID when you want to apply style to an inline chunk of code.

```
<span class = "highlight">This text</span> will be highlighted.
```

### <em>

The em tag is used for standard emphasis. By default, <em> italicizes text, but you can use CSS to make any other type of emphasis you wish.

### <strong>

This tag represents strong emphasis. By default, it is bold, but you can modify the formatting with CSS.

## FORMS

Forms are the standard user input mechanism in HTML / XHTML. You will need another language like JavaScript or PHP to read the contents of the form elements and act upon them.

### Form Structure

A number of tags are used to describe the structure of the form. Begin by looking over a basic form:

```
<form action = "">
  <fieldset>
    <legend>My form</legend>
    <label for = "txtName">Name</label>
    <input type = "text"
      id = "txtName" />
    <button type = "button"
      Onclick = "doSomething()">
      Do something
    </button>
  </fieldset>
</form>
```

#### Form

The <form></form> pair describes the form. In XHTML strict, you must indicate the form's action property. This is typically the server-side program that will read the form. If there is no such program, you can set the action to null ("") The method attribute is used to determine whether the data is sent through the get or post mechanism.

#### Fieldset

Most form elements are inline tags, and must be encased in a block element. The fieldset is designed exactly for this purpose. Its default appearance draws a box around the form. You can have multiple fieldsets inside a single form.

### Legend

You can add a legend inside a fieldset. This describes the purpose of the fieldset.

### Label

A label is a special inline element that describes a particular field. A label can be paired with an input element by putting that element's ID in the label's for attribute.

### Input

The input element is a general purpose inline element. It is meant to be used inside a form, and it is the basis for several types of more specific input. The subtype is indicated by the type attribute. Input elements usually include an id attribute (used for CSS and JavaScript identification) and / or a name attribute (used in server-side programming.) The same element can have both a name and an id.

### Text

This element allows a single line of text input:

```
<input type = "text"
  id = "myText"
  name = "myText" />
```

### Password

Passwords display just like textboxes, except rather than showing the text as it is typed, an asterisk appears for each letter. Note that the data is not encoded in any meaningful way. Typing text into a password field is still entirely insecure.

```
<input type = "password"
  id = "myPWD" />
```

### Radio Button

Radio buttons are used in a group. Only one element of a radio group can be selected at a time. Give all members of a radio group the same name value to indicate they are part of a group.

```
<input type = "radio"
  name = "radSize"
  value = "small"
  id = "radSmall"
  selected = "selected" />
<label for = "radSmall">Small</label>
<input type = "radio"
  name = "radSize"
  value = "large"
  id = "radLarge" />
<label for = "radLarge">Large</label>
```

Attaching a label to a radio button means the user can activate the button by clicking on the corresponding label. For best results, use the selected attribute to force one radio button to be the default.

### Checkbox

Checkboxes are much like radio buttons, but they are independent. Like radio buttons, they can be associated with a label.

```
<input type = "checkbox"
  id = "chkFries" />
<label for = "chkFries">Would you like fries with that?</label>
```

### Hidden

Hidden fields hold data that is not visible to the user (although it is still visible in the code) It is primarily used to preserve state in server-side programs.

```
<input type = "hidden"
  name = "txtHidden"
  value = "recipe for secret sauce" />
```

Note that the data is still not protected in any meaningful way.

**Button**

Buttons are used to signal user input. Buttons can be created through the input tag:

```
<input type = "button"
value = "launch the missiles"
onclick = "launchMissiles()" />
```

This will create a button with the caption "launch the missiles." When the button is clicked, the page will attempt to run a JavaScript function called "launchMissiles()" Standard buttons are usually used with JavaScript code on the client. The same button can also be created with this alternate format:

```
<button type = "button"
Onclick = "launchMissiles()">
 Launch the missiles
</button>
```

This second form is preferred because buttons often require different CSS styles than other input elements. This second form also allows an <img> tag to be placed inside the button, making the image act as the button.

**Reset**

The reset button automatically resets all elements in its form to their default values. It doesn't require any other attributes.

```
<input type = "reset" />
<button type = "reset">
 Reset
</button>
```

**Select / option**

Drop-down lists can be created through the select / option mechanism. The select tag creates the overall structure, which is populated by option elements.

```
<select id = "selColor">
 <option value = "#000000">black</option>
 <option value = "#FF0000">red</option>
 <option value = "#FFFFFF">white</option>
</select>
```

The select has an id (for client-side code) or name (for server-side code) identifier. It contains a number of options. Each option has a value which will be returned to the program. The text between <option> and </option> is the value displayed to the user. In some cases (as in this example) the value displayed to the user is not the same as the value used by programs.

**Multiple Selections**

You can also create a multi-line selection with the select and option tags:

```
<select id = "selColor"
size = "3"
multiple = "multiple">
 <option value = "#000000">black</option>
 <option value = "#FF0000">red</option>
 <option value = "#FFFFFF">white</option>
</select>
```

**DEPRECATED FORMATTING TAGS**

Certain tags common in older forms of HTML are no longer recommended as CSS provides much better alternatives.

**Font**

The font tag was used to set font color, family (typeface) and

size. Numerous CSS attributes replace this capability with much more flexible alternatives. See the CSS refcard for details.

**I (italics)**

HTML code should indicate the level of emphasis rather than the particular stylistic implications. Italicizing should be done through CSS. The <em> tag represents emphasized text. It produces italic output unless the style is changed to something else. The <i> tag is no longer necessary and is not recommended. Add font-style: italic to the style of any element that should be italicized.

**B (bold)**

Like italics, boldfacing is considered a style consideration. Use the <strong> tag to denote any text that should be strongly emphasized. By default, this will result in boldfacing the enclosed text. You can add bold emphasis to any style with the font-weight: bold attribute in CSS.

**DEPRECATED TECHNIQUES**

In addition to the deprecated tags, there are also techniques which were once common in HTML that are no longer recommended.

**Frames**

Frames have been used as a layout mechanism and as a technique for keeping one part of the page static while dynamically loading other parts of the page in separate frames. Use of frames has proven to cause major usability problems. Layout is better handled through CSS techniques, and dynamic page generation is frequently performed through server-side manipulation or AJAX.

**Table-based design**

Before CSS became widespread, HTML did not have adequate page formatting support. Clever designers used tables to provide an adequate form of page layout. CSS provides a much more flexible and powerful form of layout than tables, and keeps the HTML code largely separated from the styling markup.

**HTML ENTITIES**

Sometimes you need to display a special character in a web page. HTML has a set of special characters for exactly this purpose. Each of these entities begins with the ampersand(&) followed by a code and a semicolon.

Character	Name	Code	Note
	Non-breaking space	&nbsp;	Adds white space
<	Less than	&lt;	Used to display HTML code or mathematics
>	Greater than	&gt;	Used to display HTML code or mathematics
&	Ampersand	&amp;	If you're not displaying an entity but really want the & symbol
©	Copyright	&copy;	Copyright symbol
®	Registered trademark	&reg;	Registered trademark

Numerous other HTML entities are available and can be found in online resources like w3schools.

**HTML 5 / CSS 3 PREVIEW**

New technologies are on the horizon. Firefox 3.5 now has support for significant new HTML 5 features, and CSS 3 is not far behind. While the following should still be considered experimental, they are likely to become very important tools in the next few years. Firefox 3.5, Safari 4 (and a few other recent browsers) support the following new features:

**Audio and video tags**

Finally the browsers have direct support for audio and video without plugin technology. These tags work much like the img tag.

```
<video src = "myVideo.ogg" autoplay>
  Your browser does not support the video tag.
</video>
<audio src = "myAudio.ogg" controls>
  Your browsers does not support the audio tag
</audio>
```

The HTML 5 standard currently supports Ogg Theora video, Ogg Vorbis audio, and wav audio. The Ogg formats are open-source alternatives to proprietary formats, and plenty of free

tools convert from more standard video formats to Ogg. The autoplay option causes the element to play automatically. The controls element places controls directly into the page.

The code between the beginning and ending tag will execute if the browser cannot process the audio or video tag. You can place alternate code here for embedding alternate versions (Flash, for example)

**The Canvas tag**

The canvas tag offers a region of the page that can be drawn upon (usually with Javascript.) This creates the possibility of real interactive graphics without requiring plugins like Flash.

**Font Face**

This is actually a CSS improvement, but it's much needed. It allows you to define a font-face in CSS and include a ttf font file from the server. You can then use this font face in your ordinary CSS and use the downloaded font. If this becomes a standard, we will finally have access to reliable downloadable fonts on the web, which will usher in web typography at long last.

**ABOUT THE AUTHOR**



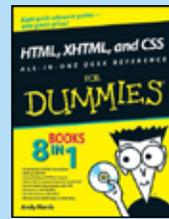
**Andy Harris**

I am a lecturer in computer science at Indiana University / Purdue University - Indianapolis. I've been interested in computing since the early eighties, when my brother and I took the money we were saving for a car and blew it on a TRS-80 model 1 with the built-in Japanese character set. My favorite part about writing is hearing from readers. It's great when somebody sends me a link to a game or project they've written using one of my books. I'd love to get a line from you about one of my books, what

you'd like to see me work on next, or just to say hi. I hope my writing feels like a conversation, and I'm looking forward to hearing from your side of the talk.

Andy's Website: <http://www.aharrisbooks.net>

**RECOMMENDED BOOK**

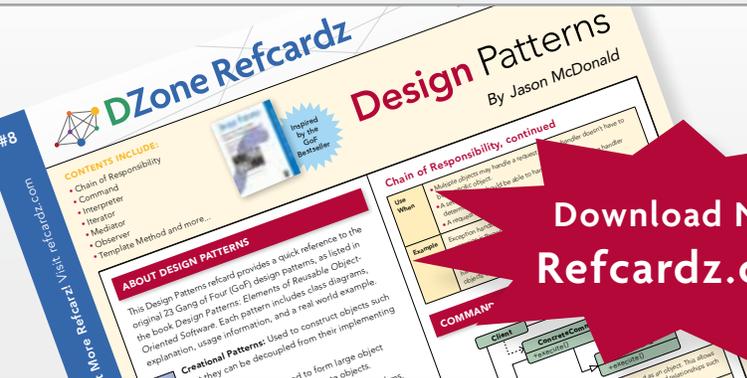


You don't need expensive or complicated software or a super-powerful computer to build a Web site that does all sorts of amazing things. All you need is a text editor and the clear, step-by-step guidance you'll find in HTML, XHTML, and CSS All-In-One Desk Reference For Dummies.

**BUY NOW**

[books.dzone.com/books/html-xhtml-css-dummies](http://books.dzone.com/books/html-xhtml-css-dummies)

**Professional Cheat Sheets You Can Trust**



**Download Now Refcardz.com**

*"Exactly what busy developers need: simple, short, and to the point."*

James Ward, Adobe Systems

**Upcoming Titles**

- Adobe ColdFusion
- Selenium
- Virtualization
- ASP.NET MVC Framework
- Oracle Berkeley DB
- Java Performance Tuning
- Eclipse Plug-In Development

**Most Popular**

- Spring Configuration
- jQuery Selectors
- Windows Powershell
- Dependency Injection with EJB 3
- Netbeans IDE JavaEditor
- Getting Started with Eclipse
- Very First Steps in Flex



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more.

"DZone is a developer's dream," says PC Magazine.

DZone, Inc.  
140 Preston Executive Dr.  
Suite 100  
Cary, NC 27513  
888.678.0399  
919.678.0300

Refcardz Feedback Welcome  
[refcardz@dzone.com](mailto:refcardz@dzone.com)

Sponsorship Opportunities  
[sales@dzone.com](mailto:sales@dzone.com)



\$7.95