Speed Up Your Site: Web Optimization

0735713243

Andrew B. King

Copyright© 2003 by New Riders Publishing

Warning and Disclaimer: Every effort has been made to make this book as complete and accurate as possible, but no warranty or fitness is implied. The information is provided on an as-is basis. The authors and New Riders Publishing shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

Misprint	Correction
Page xv Stuart Robinson	Stuart Robertson
Page xv Ralph Engelschall	Ralf Engelschall
Page 50 Here's an abbreviated log file of Elivad.com, which uses graphic rollovers after loading this page:	Here's an abbreviated log file of Elivad.com, after loading this unoptimized page:
Page 54 `font-size   1.1 em`	`font-size 1.1em`
Page 57 `table.tr.right{text-align:right;}`	`tr.right{text-align:right;}`
Page 57 `<img src="t.gif" width=1 height=1>`	`<img src=t.gif width=1 height=1>`
Page 57 `<table width=100%>`	`<img src=images/t.gif width=10% height=10%>`
Page 57 Values with spaces, symbols, or links require quotes in HTML. For example: `<a href="/index.html">`	Values with any other characters such as spaces, slashes, or ampersands require quotes in HTML. For example: `<a href="/index.html">...</a>`
Page 63 Do this: `<img src-"t.gif" alt=""width="1" height="1">`	`<img src-"t.gif" alt="" width="1" height="1">`
Page 87 The GET method is more efficient than POST because it takes one less trip to the server. When security is not paramount, ising GET (or non-parsed headers) can mean faster form processing for your users.	The GET method is less secure than POST because the form's data is appended onto the end of the action URI and saved in the server's logs. You can use non-parsed header scripts for faster CGI processing, including scripts that process forms.
Page 95 For example, the home page of PopularMechanics.com ca be compressed from 138,548 to 21K using gzip compression (see Table 4.1).	For example, the home page of PopularMechanics.com ca be compressed from 138,548 to 21,743 bytes using gzip compression (see Table 4.1).
Page 96-Table 4.1 Table column head: PopularMechanics.com (HTML)	HTML Size (Bytes)
Page 101 `do you change the font site-wide?`	`do you change the font site-wide?</font></p>`
Page 101 `"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd ">`	`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
Page 103 `"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd ">`	`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
Page 106 `"http://www.w3.org/TR/xhtml1/DTD/xhtml1-`	`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

strict.dtd ">	
Page 115 Some existing HTML browsers don’t support the `id` attribute; therefore, you should supply both the `id` and `name` attributes to ensure forward and backward compatibility (for example, `<a id=“review” name=“review”>…</a>`).	Some existing HTML browsers don’t support the `id` attribute; therefore, you should supply both the `id` and `name` attributes to ensure forward and backward compatibility (for example, `<a id="review" name="review">…</a>`).
Page 118 `specified paragraph text goes here.</p>`	`specified paragraph text goes here.</font></p>`
Page 119 (2 occurrences on page) `"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd ">`	`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
Page 134 `<option value="1903 ">1903`	`<option value="1903">1903`
Page 140 `"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd ">`	`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
Page 150 There are six types of shorthand properties defined in CSS; `font`, `background`, `margin`, `border`, `padding`, and `list`.	There are six types of shorthand properties defined in CSS; `font`, `background`, `margin`, `border`, `padding`, and `list-style`.
Page 150 `{ font: weight style variant size/line-height font-name(s) inherit }`	`{ font: weight style variant size/line-height font-name(s) | inherit }`
Page 151 `{ background: color image repeat attachment position inherit }`	`{ background: color image repeat attachment position | inherit }`
Page 159-Figure 7.2	Should be identical to Figure 7.1.

Page 159
`ul li { list-style:`
`url(http://domain.gif/b.gif) square outside}`
	`ul li { list-style:`
	`url(http://domain.com/b.gif) square outside}`

Page 159
`ul { list-style:`
`url(http://domain.gif/b.gif) square }`
	`ul { list-style:`
	`url(http://domain.com/b.gif) square }`

Page 163 Macs typically have 72 dpi displays, although this isn’t guaranteed because font sizing percentage values are relative to another value and work like the `em` unit.	Macs typically have 72 ppi displays, although this isn’t guaranteed because font sizing percentage values are relative to another value and work like the `em` unit.
Pages 171 and 177 Stuart Robinson	Stuart Roberston
Page 197 `"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd ">`	`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
Page 205 JSCruncher from Hoard’s DOMAPI project (`http://www.domapi.com/`)	JSCruncher from the DOMAPI project (`http://www.domapi.com/`)
Page 217 Reduce the number of dots (`object.property`) and brackets (`object[“property”]`) in your program by caching frequently used objects and properties.	Reduce the number of dots (`object.property`) and brackets (`object["property"]`) in your program by caching frequently used objects and properties.
Page 228 `var serialData=new;`	`var serialData=new`

Page 232- Listing 10.14
`function loopDoWhileReverse3() {`
`var i=iter;`
`do`
`{`
`    // do something here`
`}`
`while (--i);`
`}`
	`function loopDoWhileReverse3() {`
	`var i=iter;`
	`if (i>0) { // make sure i is positive here`
	`    do`
	`    {`
	`        // do something here`
	`    }`
	`    while (--i); // i must be greater than 0 here`
	`    }`
	`}`

Page 232 – Table 10.2	Row 1: 2372, 2299, 1181, 1082, 706, 581

Change to number in table. Row 1: 2022, 1958, 1018, 932, 609, 504 Row 2: 0.0040, 0.0039, 0.0020, 0.0018, 0.0012, 0.0010	Row 2: .0047, .0046, .0024, .0022, .0014, .0012

Page 236 – Listing10.18
`n = parseInt(iterations / 8);`
`     do`
`     {`
`         testVal++;`
`         testVal++;`
`         testVal++;`
`         testVal++;`
`         testVal++;`
`         testVal++;`
`         testVal++;`
`         testVal++;`
`     }`
`     while (--n);`
`}`
	`n = parseInt(iterations / 8);`
	`     if (n>0) { // check to make sure n is positive here`
	`         do`
	`         {`
	`             testVal++;`
	`             testVal++;`
	`             testVal++;`
	`             testVal++;`
	`             testVal++;`
	`             testVal++;`
	`             testVal++;`
	`             testVal++;`
	`         }`
	`         while (--n); // n must be greater than 0 here`
	`     }`
	`}`

Page 236 – Table 10.3 Change to number in table. Row 1: 1437, 775, 493, 469 Row 2: 0.00287, 0.00155, 0.00099, 0.00094	Row 1: 1628, 890, 558, 540 Row 2: .00326, .00178, .00112, .00108
Page 236 – Table 10.4 Change to number in table. Row 1: 925, 661, 576, 533, 509, 490, 482 Row 2: 0.00184, 0.00132, 0.00115, 0.00106, 0.00097, 0.00096	Row 1: 1056, 765, 662, 616, 585, 567, 551 Row 2: .00211, .00153, .00132, .00123, .00117, .00113, .00110
Page 237 – Table 10.5 Change to number in table. Row 1: 469, 467, 457, 453, 439, 437, 433, 433 Row 2: 0.00093, 0.00093, 0.00091, 0.00090, 0.00087, 0.00087, 0.00086, 0.00086	Row 1: 540, 533, 525, 520, 515, 512, 510, 507 Row 2: .00108, .00107, .00105, .00104, .00103, 00102, .00102, .00101
Page 237 Around seven statements, the time is cut by two-thirds.	Around eight statements, the time is cut by two-thirds.
Page 245 Manually packing to abbreviate variable and function names saves 52 percent, reducing from 55,573 to 26,870 bytes.	Manually packing to abbreviate variable and function names saves 52 percent, reducing the combined DOM file from 55,573 to 26,870 bytes.
Page 245-Table 11.3 Table column head: Size	Size in Bytes
Page 246 GZIP compressing the original file save over 80 percent in file size (from 55,573 to 10,655bytes).	GZIP compressing the original combined DOM file save over 80 percent in file size (from 55,573 to 10,655bytes).
Page 304-Table 13.1 Table column head: QDX	QDM
Page 305 This helps the source encode as efficiently as possible and look good at smaller image sizes.	This helps the codec as efficiently as possible, make your movie look good at smaller image sizes.
Page 330 ▪ **Keyframes**—“every” 90 is used.	▪ **Keyframes**—“every 90” is used.
Page 366 Add your keywords into `key` tags and attributes, and mix well.	Add your keywords into key tags and attributes, and mix well.
Page 366 `<p><b>The Original™ Search Engine Positioning Firm</b>`	`<p><b>The Original™ Search Engine Positioning Firm</b></p>`
Page 401	The `method` attribute sets the type of HTTP

The method attribute sets the HTTP method the browser uses to send the form’s data to the server. There are two methods of sending data: POST and GET. With the POST method, the browser sends the data in two steps—first it contacts the server’s action script; then it sends the data to the server in a separate transmission. The data is hidden from the user using this method. In the GET method (the default) the browser contacts the server’s action script and appends the data onto the URL, all in one transmission. GETs are more efficient than POSTs, but less secure because the data appears and can be modified in the browser’s address field.	request the browser will use to send the form’s data to the server. The two most common methods of sending data are GET and POST. With the POST method, the browser sends the data as the body of the HTTP request the data is hidden from the user using this method. In the GET method (the default) the browser contacts the server’s action script and appends the encoded data onto the URL. GETs are less secure because the data appears and can be modified in the browser’s address field. The query string is also saved in the server’s logs, where anyone who has access to the log files could see all the form data submitted.
Page 402 `my $Revelancethreshold= 50;     # relevance threshold, 0-100`	`my $Revelancethreshold_DEF = 50;     # relevance threshold, 0-100`
Page 477 Missing index entry under **Apache**	mod_include 375-377
Page 482 **Duffy, Tom, 233**	**Duff, Tom, 233**
Page 48 Missing index entry	Engelschall, Ralf 392
Page 484 Missing index entry under **graphics**	optimizing, 249-281
Page 488 Missing index entry under **multimedia**	optimizing, 297-325
Page 491 Robinson, Stuart	Robertson, Stuart

This errata sheet is intended to provide updated technical information.
Spelling and grammar misprints are updated during the reprint process,
but are not listed on this errata sheet.