

# Web Application Development with PHP 4.0

0-7357-0997-1

Tobias Ratschiller and Till Gerken

Copyright© 2001 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								
<b>CD-ROM</b> The knowledge repository application is missing on the CD-ROM.	It is downloadable from <a href="http://www.phpwizard.net/book/errata.html">http://www.phpwizard.net/book/errata.html</a> .								
<b>Page 30</b> \$event_list = calendar_get_event_list(\$required_range )	\$event_list = calendar_get_event_list(\$required_ range)								
<b>Page 35</b> <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>3</td><td>Element 3</td></tr></tbody></table>	Key	Value	3	Element 3	<table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>2</td><td>Element 3</td></tr></tbody></table>	Key	Value	2	Element 3
Key	Value								
3	Element 3								
Key	Value								
2	Element 3								
<b>Page 40</b> A call to each() has been inserted between the two loops. The second usage of each() restarts retrieving key/value pairs from the first element.	A call to reset() has been inserted between the two loops. The second usage of reset() restarts retrieving key/value pairs from the first element.								
<b>Page 48</b> Member function can be called by either using the “old” syntax of instance->member() or the “new syntax instance::member().	Member function can be called by either using the “old” syntax of instance->member() or the “new syntax classname::member().								
<b>Page 49</b> <i>Note:</i> Since PHP 4.0, constructors can only take scalar values as parameters (strings, integers, etc.), but no arrays or objects, which was still possible with PHP 3.0.	<i>Note:</i> Previously with PHP 4.0, constructors could only take scalar values as parameters (strings, integers, etc.), but no arrays or objects, which was still possible with PHP 3.0. This was a temporary bug; constructors, default values of class variables, and default values of function arguments now accept all types of arguments.								
<b>Page 71</b> while(get_encoded_pair(&\$count, &\$value))	while(get_encoded_pair(\$count, \$value))								
<b>Page 135</b> You can use serialize() and unserialize(), but note that in PHP 3.0 these functions don’t work correctly on objects (classes); class functions will be discarded.	You can use serialize() and unserialize(), but note that in PHP 3.0 these functions don’t work correctly on objects (classes); class functions will be discarded.								
<b>Page 139</b> When using public caching, the cache_expire PHP configuration directive will be taken into account.	When using public and private caching, the cache_expire PHP configuration directive will be taken into account.								
<b>Page 189</b> The concept of PHP streams was introduced with PHP 4.0 and is not available in the 3.x tree.	The concept of PHP streams was added in PHP 3.0.13 first and ported later to PHP 4.0.								
<b>Page 220</b>	The result of this would be 1. In binary:								

The result of this would be 1. In binary: 0000010	0000011
<b>Page 221</b> (Last line in “Bitwise Shifts”) = 0000010	= 0000100
<b>Page 343</b> *return_value == **parameter;	*return_value = **parameter;

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.