

"This book, written by my colleague, Andi Gutmans and two very prominent PHP developers, Stig Bakken and Derick Rethans, holds the key to unlocking the riches of PHP 5. It thoroughly covers all of the features of this new version, and is a must have for all PHP developers interested in exploring PHP 5's advanced features"

—Zeev Suraski, Co-Designer of PHP 5 and Co-Creator of the Zend Engine

BRUCE PERENS' OPEN SOURCE SERIES

PHP 5

Power Programming



- Expert PHP 5 programming techniques, *direct from PHP 5 co-creator and key developers*
- Definitive coverage of the PHP object model, design patterns, XML/SOAP integration, and more
- Solid introduction and expert insights into performance tuning, PEAR, extensions, and database access
- The definitive introduction to PHP 5's advanced features!

**Free 90-day trial
version of Zend Studio!**

ANDI GUTMANS · STIG SÆTHER BAKKEN · DERICK RETHANS
LEAD DESIGNER OF PHP 5 CREATOR OF PEAR KEY PHP CONTRIBUTOR

Contents

Foreword	xi
Preface	xiii
1 What Is New in PHP 5?	1
1.1 Introduction	1
1.2 Language Features	1
1.2.1 New Object-Oriented Model.....	1
1.2.2 New Object-Oriented Features.....	3
1.2.3 Other New Language Features	7
1.3 General PHP Changes.....	8
1.3.1 XML and Web Services	8
1.4 Other New Features in PHP 5	11
1.4.1 New Memory Manager.....	11
1.4.2 Dropped Support for Windows 95.....	11
1.5 Summary.....	11
2 PHP 5 Basic Language	13
2.1 Introduction	13
2.2 HTML Embedding.....	14
2.3 Comments	14
2.4 Variables.....	15
2.4.1 Indirect References to Variables	16
2.4.2 Managing Variables	16
2.4.3 Superglobals	18
2.5 Basic Data Types	18
2.5.1 Integers	19
2.5.2 Floating-Point Numbers	19
2.5.3 Strings.....	19
2.5.4 Booleans.....	22
2.5.5 Null	23

2.5.6 Resources	23
2.5.7 Arrays	23
2.5.8 Constants	30
2.6 Operators	31
2.6.1 Binary Operators	32
2.6.2 Assignment Operators	32
2.6.3 Comparison Operators	33
2.6.4 Logical Operators	34
2.6.5 Bitwise Operators	35
2.6.6 Unary Operators	36
2.6.7 Negation Operators	36
2.6.8 Increment/Decrement Operators	37
2.6.9 The Cast Operators	38
2.6.10 The Silence Operator	39
2.6.11 The One and Only Ternary Operator	39
2.7 Control Structures	39
2.7.1 Conditional Control Structures	39
2.7.2 Loop Control Structures	42
2.7.3 Code Inclusion Control Structures	45
2.8 Functions	48
2.8.1 User-Defined Functions	49
2.8.2 Function Scope	49
2.8.3 Returning Values By Value	50
2.8.4 Returning Values By Reference	51
2.8.5 Declaring Function Parameters	52
2.8.6 Static Variables	53
2.9 Summary	54
3 PHP 5 OO Language	55
3.1 Introduction	55
3.2 Objects	55
3.3 Declaring a Class	57
3.4 The new Keyword and Constructors	57
3.5 Destructors	58
3.6 Accessing Methods and Properties Using the \$this Variable	59
3.6.1 public , protected , and private Properties	60
3.6.2 public , protected , and private Methods	61
3.6.3 Static Properties	62
3.6.4 Static Methods	64
3.7 Class Constants	65
3.8 Cloning Objects	66
3.9 Polymorphism	67
3.10 parent:: and self::	70
3.11 instanceof Operator	71

3.12 Abstract Methods and Classes	72
3.13 Interfaces	73
3.14 Inheritance of Interfaces	75
3.15 final Methods	75
3.16 final Classes	76
3.17 __toString() Method	76
3.18 Exception Handling	77
3.19 __autoload()	80
3.20 Class Type Hints in Function Parameters	82
3.21 Summary	83
4 PHP 5 Advanced OOP and Design Patterns	85
4.1 Introduction	85
4.2 Overloading Capabilities	85
4.2.1 Property and Method Overloading	85
4.2.2 Overloading the Array Access Syntax	88
4.3 Iterators	89
4.4 Design Patterns	94
4.4.1 Strategy Pattern	95
4.4.2 Singleton Pattern	97
4.4.3 Factory Pattern	98
4.4.4 Observer Pattern	101
4.5 Reflection	103
4.5.1 Introduction	103
4.5.2 Reflection API	103
4.5.3 Reflection Examples	106
4.5.4 Implementing the Delegation Pattern Using Reflection	107
4.6 Summary	109
5 How to Write a Web Application with PHP	111
5.1 Introduction	111
5.2 Embedding into HTML	112
5.3 User Input	114
5.4 Safe-Handling User Input	117
5.4.1 Common Mistakes	117
5.5 Techniques to Make Scripts "Safe"	120
5.5.1 Input Validation	120
5.5.2 HMAC Verification	122
5.5.3 PEAR::Crypt_HMAC	124
5.5.4 Input Filter	127
5.5.5 Working with Passwords	127
5.5.6 Error Handling	129
5.6 Cookies	131
5.7 Sessions	134

5.8 File Uploads	137
5.8.1 Handling the Incoming Uploaded File	138
5.9 Architecture	143
5.9.1 One Script Serves All	143
5.9.2 One Script per Function	144
5.9.3 Separating Logic from Layout	144
5.10 Summary	146
6 Databases with PHP 5	149
6.1 Introduction	149
6.2 MySQL	149
6.2.1 MySQL Strengths and Weaknesses	150
6.2.2 PHP Interface	150
6.2.3 Example Data	151
6.2.4 Connections	151
6.2.5 Buffered Versus Unbuffered Queries	153
6.2.6 Queries	154
6.2.7 Multi Statements	155
6.2.8 Fetching Modes	156
6.2.9 Prepared Statements	156
6.2.10 BLOB Handling	158
6.3 SQLite	160
6.3.1 SQLite Strengths and Weaknesses	160
6.3.2 Best Areas of Use	161
6.3.3 PHP Interface	162
6.4 PEAR DB	176
6.4.1 Obtaining PEAR DB	176
6.4.2 Pros and Cons of Database Abstraction	177
6.4.3 Which Features Are Abstracted?	177
6.4.4 Database Connections	178
6.4.5 Executing Queries	180
6.4.6 Fetching Results	182
6.4.7 Sequences	184
6.4.8 Portability Features	185
6.4.9 Abstracted Errors	186
6.4.10 Convenience Methods	188
6.5 Summary	190
7 Error Handling	191
7.1 Introduction	191
7.2 Types of Errors	192
7.2.1 Programming Errors	192
7.2.2 Undefined Symbols	194
7.2.3 Portability Errors	197

7.2.4 Runtime Errors.....	201
7.2.5 PHP Errors	201
7.3 PEAR Errors	206
7.3.1 The PEAR_Error Class	209
7.3.2 Handling PEAR Errors	212
7.3.3 PEAR Error Modes.....	213
7.3.4 Graceful Handling	213
7.4 Exceptions	216
7.4.1 What Are Exceptions?	216
7.4.2 try, catch, and throw	216
7.5 Summary	218
8 XML with PHP 5.....	219
8.1 Introduction	219
8.2 Vocabulary	220
8.3 Parsing XML	222
8.3.1 SAX	222
8.3.2 DOM	226
8.4 SimpleXML	231
8.4.1 Creating a SimpleXML Object.....	232
8.4.2 Browsing SimpleXML Objects	233
8.4.3 Storing SimpleXML Objects	234
8.5 PEAR	234
8.5.1 XML_Tree	235
8.5.2 XML_RSS.....	236
8.6 Converting XML	239
8.6.1 XSLT	239
8.7 Communicating with XML.....	244
8.7.1 XML-RPC.....	244
8.7.2 SOAP	252
8.8 Summary	259
9 Mainstream Extensions	261
9.1 Introduction	261
9.2 Files and Streams	261
9.2.1 File Access	262
9.2.2 Program Input/Output.....	264
9.2.3 Input/Output Streams.....	267
9.2.4 Compression Streams	268
9.2.5 User Streams	270
9.2.6 URL Streams	271
9.2.7 Locking	276
9.2.8 Renaming and Removing Files	277
9.2.9 Temporary Files	278

9.3 Regular Expressions	279
9.3.1 Syntax	279
9.3.2 Functions.....	293
9.4 Date Handling.....	301
9.4.1 Retrieving Date and Time Information.....	301
9.4.2 Formatting Date and Time	305
9.4.3 Parsing Date Formats	313
9.5 Graphics Manipulation with GD.....	314
9.5.1 Case 1: Bot-Proof Submission Forms	315
9.5.2 Case 2: Bar Chart	320
9.5.3 Exif	326
9.6 Multi-Byte Strings and Character Sets.....	329
9.6.1 Character Set Conversions.....	330
9.6.2 Extra Functions Dealing with Multi-Byte Character Sets	335
9.6.3 Locales.....	340
9.7 Summary	343
10 Using PEAR.....	345
10.1 Introduction.....	345
10.2 PEAR Concepts	346
10.2.1 Packages.....	346
10.2.2 Releases.....	346
10.2.3 Version Numbers	347
10.3 Obtaining PEAR.....	349
10.3.1 Installing with UNIX / Linux PHP Distribution	350
10.3.2 Installing with PHP Windows Installer	351
10.3.3 go-pear.org	351
10.4 Installing Packages.....	354
10.4.1 Using the pear Command	354
10.5 Configuration Parameters.....	358
10.6 PEAR Commands.....	364
10.6.1 pear install	364
10.6.2 pear list	368
10.6.3 pear info	369
10.6.4 pear list-all	370
10.6.5 pear list-upgrades	370
10.6.6 pear upgrade	371
10.6.7 pear upgrade-all	372
10.6.8 pear uninstall	373
10.6.9 pear search	373
10.6.10 pear remote-list	374
10.6.11 pear remote-info	375
10.6.12 pear download	375
10.6.13 pear config-get	376

10.6.14	pear config-set	376
10.6.15	pear config-show	376
10.6.16	Shortcuts	377
10.7	Installer Front-Ends	378
10.7.1	CLI (Command Line Interface) Installer	378
10.7.2	Gtk Installer	378
10.8	Summary	381
11	Important PEAR Packages	383
11.1	Introduction	383
11.2	Database Queries	383
11.3	Template Systems	383
11.3.1	Template Terminology	384
11.3.2	HTML_Template_IT	384
11.3.3	HTML_Template_Flexy	387
11.4	Authentication	392
11.4.1	Overview	392
11.4.2	Example: Auth with Password File	393
11.4.3	Example: Auth with DB and User Data	394
11.4.4	Auth Security Considerations	396
11.4.5	Auth Scalability Considerations	397
11.4.6	Auth Summary	398
11.5	Form Handling	398
11.5.1	HTML_QuickForm	398
11.5.2	Example: Login Form	399
11.5.3	Receiving Data	399
11.6	Caching	399
11.6.1	Cache_Lite	399
11.7	Summary	401
12	Building PEAR Components	403
12.1	Introduction	403
12.2	PEAR Standards	403
12.2.1	Symbol Naming	403
12.2.2	Indentation	406
12.3	Release Versioning	408
12.4	CLI Environment	408
12.5	Fundamentals	410
12.5.1	When and How to Include Files	410
12.5.2	Error Handling	411
12.6	Building Packages	411
12.6.1	PEAR Example: HelloWorld	411
12.6.2	Building the Tarball	414
12.6.3	Verification	414
12.6.4	Regression Tests	416

12.7 The package.xml Format	416
12.7.1 Package Information	417
12.7.2 Release Information	419
12.8 Dependencies.....	423
12.8.1 Element: <code><deps></code>	423
12.8.2 Element: <code><dep></code>	423
12.8.3 Dependency Types	424
12.8.4 Reasons to Avoid Dependencies	425
12.8.5 Optional Dependencies.....	426
12.8.6 Some Examples	426
12.9 String Substitutions.....	427
12.9.1 Element: <code><replace></code>	427
12.9.2 Examples.....	427
12.10 Including C Code.....	428
12.10.1 Element: <code><configureoptions></code>	428
12.10.2 Element: <code><configureoption></code>	428
12.11 Releasing Packages.....	428
12.12 The PEAR Release Process.....	429
12.13 Packaging	430
12.13.1 Source Analysis.....	430
12.13.2 MD5 Checksum Generation.....	430
12.13.3 Package.xml Update	431
12.13.4 Tarball Creation	431
12.14 Uploading.....	432
12.14.1 Upload Release	432
12.14.2 Finished!.....	432
12.15 Summary	432
13 Making the Move	433
13.1 Introduction.....	433
13.2 The Object Model	433
13.3 Passing Objects to Functions.....	433
13.4 Compatibility Mode.....	435
13.4.1 Casting Objects.....	435
13.4.2 Comparing Objects	436
13.5 Other Changes	437
13.5.1 Assigning to <code>\$this</code>	437
13.5.2 <code>get_class</code>	440
13.6 <code>E_STRICT</code>	441
13.6.1 Automatically Creating Objects	441
13.6.2 <code>var</code> and <code>public</code>	441
13.6.3 Constructors.....	442
13.6.4 Inherited Methods	442
13.6.5 Define Classes Before Usage.....	443

13.7 Other Compatibility Problems	443
13.7.1 Command-Line Interface	443
13.7.2 Comment Tokens	443
13.7.3 MySQL	445
13.8 Changes in Functions	445
13.8.1 <code>array_merge()</code>	445
13.8.2 <code>strrpos()</code> and <code>stripos()</code>	446
13.9 Summary	447
14 Performance	449
14.1 Introduction	449
14.2 Design for Performance	449
14.2.1 PHP Design Tip #1: Beware of State	450
14.2.2 PHP Design Tip #2: Cache!	451
14.2.3 PHP Design Tip #3: Do Not Over Design!	456
14.3 Benchmarking	457
14.3.1 Using ApacheBench	457
14.3.2 Using Siege	458
14.3.3 Testing Versus Real Traffic	459
14.4 Profiling with Zend Studio's Profiler	459
14.5 Profiling with APD	461
14.5.1 Installing APD	461
14.5.2 Analyzing Trace Data	462
14.6 Profiling with Xdebug	465
14.6.1 Installing Xdebug	466
14.6.2 Tracing Script Execution	466
14.6.3 Using KCacheGrind	468
14.7 Using APC (Advanced PHP Cache)	470
14.8 Using ZPS (Zend Performance Suite)	470
14.8.1 Automatic Optimization	471
14.8.2 Compiled Code Caching	472
14.8.3 Dynamic Content Caching	473
14.8.4 Content Compression	476
14.9 Optimizing Code	477
14.9.1 Micro-Benchmarks	477
14.9.2 Rewrite in C	479
14.9.3 OO Versus Procedural Code	480
14.10 Summary	481
15 An Introduction to Writing PHP Extensions	483
15.1 Introduction	483
15.2 Quickstart	484
15.2.1 Memory Management	489
15.2.2 Returning Values from PHP Functions	490
15.2.3 Completing <code>self_concat()</code>	490
15.2.4 Summary of Example	492
15.2.5 Wrapping Third-Party Extensions	492

15.2.6 Global Variables	501
15.2.7 Adding Custom INI Directives.....	503
15.2.8 Thread-Safe Resource Manager Macros.....	504
15.3 Summary	505
16 PHP Shell Scripting.....	507
16.1 Introduction.....	507
16.2 PHP CLI Shell Scripts	508
16.2.1 How CLI Differs From CGI.....	508
16.2.2 The Shell-Scripting Environment.....	510
16.2.3 Parsing Command-Line Options	512
16.2.4 Good Practices.....	515
16.2.5 Process Control	516
16.2.6 Examples.....	520
16.3 Summary	526
A PEAR and PECL Package Index.....	527
A.1 Authentication	527
A.2 Benchmarking	530
A.3 Caching	530
A.4 Configuration	531
A.5 Console	531
A.6 Database	533
A.7 Date and Time	542
A.8 Encryption	543
A.9 File Formats.....	545
A.10 File System.....	548
A.11 Gtk Components.....	550
A.12 HTML.....	550
A.13 HTTP.....	561
A.14 Images.....	563
A.15 Internationalization	566
A.16 Logging.....	568
A.17 Mail	569
A.18 Math	571
A.19 Networking	574
A.20 Numbers.....	584
A.21 Payment.....	585
A.22 PEAR.....	587
A.23 PHP	588
A.24 Processing	594
A.25 Science.....	594
A.26 Streams.....	595
A.27 Structures	596
A.28 System.....	598
A.29 Text.....	599

A.30 Tools and Utilities	600
A.31 Web Services.....	603
A.32 XML	604
B phpDocumentor Format Reference	613
B.1 Introduction.....	613
B.2 Documentation Comments	613
B.3 Tag Reference	615
B.3.1 abstract	615
B.3.2 access	616
B.3.3 author	617
B.3.4 category	618
B.3.5 copyright	618
B.3.6 deprecated	618
B.3.7 example	619
B.3.8 filesource	620
B.3.9 final	620
B.3.10 global	621
B.3.11 ignore	622
B.3.12 inheritdoc (inline)	622
B.3.13 internal, internal (inline)	622
B.3.14 licence	623
B.3.15 link	623
B.3.16 link (inline)	623
B.3.17 name	624
B.3.18 package	624
B.3.19 param	626
B.3.20 return	627
B.3.21 see	627
B.3.22 since	628
B.3.23 static	628
B.3.24 staticvar	629
B.3.25 subpackage	629
B.3.26 todo	630
B.3.27 uses	630
B.3.28 var	631
B.3.29 version	631
B.4 Tag Table.....	632
B.5 Using the phpDocumentor Tool	633
C Zend Studio Quick Start Guide	643
C.1 Version 3.5.x	643
C.2 About the Zend Studio Client Quick Start Guide.....	643
C.3 About Zend	643
C.4 Zend Studio Client: Overview	644

C.4.1 Studio Components	644
C.4.2 Client Server Configuration	645
C.4.3 Installation and Registration	645
C.5 Editing a File	647
C.5.1 Editing a File	647
C.6 Working with Projects	648
C.6.1 Advantages of Working with Projects	648
C.6.2 How to Create a Project	648
C.7 Running the Debugger	648
C.7.1 Internal Debugger	649
C.7.2 Remote Debugger	649
C.7.3 Debug URL	650
C.8 Configure Studio Server for Remote Debugger and Profiling	650
C.9 Running the Profiler	651
C.10 Product Support	652
C.10.1 Getting Support	653
C.11 Main Features	653
Index	655