Preface

"The best security against revolution is in constant correction of abuses and the introduction of needed improvements. It is the neglect of timely repair that makes rebuilding necessary."—Richard Whately

In the Beginning

It was eight years ago, when Rasmus Lerdorf first started developing PHP/FI. He could not have imagined that his creation would eventually lead to the development of PHP as we know it today, which is being used by millions of people. The first version of "PHP/FI," called **Personal Homepage Tools/ Form Interpreter,** was a collection of Perl scripts in 1995.[1] One of the basic features was a Perl-like language for handling form submissions, but it lacked many common useful language features, such as `for` loops.

[1] http://groups.google.com/groups?selm=3r7pgp$aa1@ionews.io.org.

PHP/FI 2

A rewrite came with PHP/FI 2[2] in 1997, but at that time the development was almost solely handled by Rasmus. After its release in November of that year, Andi Gutmans and Zeev Suraski bumped into PHP/FI while looking for a language to develop an e-commerce solution as a university project. They discovered that PHP/FI was not quite as powerful as it seemed, and its language was lacking many common features. One of the most interesting aspects included the way `while` loops were implemented. The hand-crafted lexical scanner would go through the script and when it hit the while keyword it would remember its position in the file. At the end of the loop, the file pointer sought back to the saved position, and the whole loop was reread and re-executed.

PHP 3

Zeev and Andi decided to completely rewrite the scripting language. They then teamed up with Rasmus to release PHP 3, and along also came a new name: PHP: Hypertext Preprocessor, to emphasize that PHP was a different product and not only suitable for personal use. Zeev and Andi had also designed and implemented a new extension API. This new API made it possible to easily support additional extensions for performing tasks such as accessing databases, spell checkers and other technologies, which attracted many developers who were not part of the "core" group to join and contribute to the PHP project. At the time of PHP 3's release[3] in June 1998, the estimated PHP installed base consisted of about 50,000 domains. PHP 3 sparked the beginning of PHP's real breakthrough, and was the first version to have an installed base of more than one million domains.

PHP 4

In late 1998, Zeev and Andi looked back at their work in PHP 3 and felt they could have written the scripting language even better, so they started yet another rewrite. While PHP 3 still continuously parsed the scripts while executing them, PHP 4 came with a new paradigm of "compile first, execute later." The compilation step does not compile PHP scripts into machine code; it instead compiles them into byte code, which is then executed by the **Zend Engine** (Zend stands for **Ze**ev & A**nd**i), the new heart of PHP 4. Because of this new way of executing scripts, the performance of PHP 4 was much better than that of PHP 3, with only a small amount of backward compatibility breakage[4]. Among other improvements was an improved extension API for better run-time performance, a web server abstraction layer allowing PHP 4 to run on most popular web servers, and lots more. PHP 4 was officially released on May 22, 2002, and today its installed base has surpassed 15 million domains.

[2] http://groups.google.com/groups?selm=Dn1JM9.61t%40gpu.utcc.utoronto.ca.

[3] http://groups.google.com/groups?selm=Pine.WNT.3.96.980606130654.-317675I-100000%40shell.lerdorf.on.ca.

[4] http://www.php.net/manual/en/migration4.php.

In PHP 3, the minor version number (the middle digit) was never used, and all versions were numbered as 3.0.x. This changed in PHP 4, and the minor version number was used to denote important changes in the language. The first important change came in PHP 4.1.0,[5] which introduced **superglobals** such as `$_GET` and `$_POST`. Superglobals can be accessed from within functions without having to use the `global` keyword. This feature was added in order to allow the `register_globals` INI option to be turned off. `register_globals` is a feature in PHP which automatically converts input variables like `"?foo=bar" in http://php.net/?foo=bar` to a PHP variable called `$foo`. Because many people do not check input variables properly, many applications had security holes, which made it quite easy to circumvent security and authentication code.

With the new superglobals in place, on April 22, 2002, PHP 4.2.0 was released with the `register_globals` turned off by default. PHP 4.3.0, the last significant PHP 4 version, was released on December 27, 2002. This version introduced the **Command Line Interface** (CLI), a revamped file and network I/O layer (called **streams**), and a bundled GD library. Although most of those additions have no real effect on end users, the major version was bumped due to the major changes in PHP's core.

PHP 5

Soon after, the demand for more common object-oriented features increased immensely, and Andi came up with the idea of rewriting the objected-oriented part of the Zend Engine. Zeev and Andi wrote the "Zend Engine II: Feature Overview and Design" document[6] and jumpstarted heated discussions about PHP's future. Although the basic language has stayed the same, many features were added, dropped, and changed by the time PHP 5 matured. For example, namespaces and multiple inheritance, which were mentioned in the original document, never made it into PHP 5. Multiple inheritance was dropped in favor of interfaces, and namespaces were dropped completely. You can find a full list of new features in Chapter, "What Is New in PHP 5?"

PHP 5 is expected to maintain and even increase PHP's leadership in the web development market. Not only does it revolutionizes PHP's object-oriented support but it also contains many new features which make it the ultimate web development platform. The rewritten XML functionality in PHP 5 puts it on par with other web technologies in some areas and overtakes them in others, especially due to the new SimpleXML extension which makes it ridiculously easy to manipulate XML documents. In addition, the new SOAP, MySQLi, and variety of other extensions are significant milestones in PHP's support for additional technologies.

[5] http://www.php.net/release_4_1_0.php.

[6] http://zend.com/engine2/ZendEngine-2.0.pdf.

AUDIENCE

This book is an introduction to the advanced features new to PHP 5. It is written for PHP programmers who are making the move to PHP 5. Although Chapter 2, "PHP 5 Basic Language," contains an introduction to PHP 5 syntax, it is meant as a refresher for PHP programmers and not as a tutorial for new programmers. However, web developers with experience programming other high-level languages may indeed find that this tutorial is all they need in order to begin working effectively with PHP 5.

CHAPTER OVERVIEW

Chapter 1, "What Is New in PHP 5?" discusses the new features in PHP 5. Most of these new features deal with new object-oriented features, including small examples for each feature. It also gives an overview of the new extensions in PHP 5. Most of the topics mentioned in this chapter are explained in more detail in later chapters.

Chapter 2, "PHP 5 Basic Language," introduces the PHP syntax to those readers not familiar with PHP. All basic language constructs and variable types are explained along with simple examples to give the reader the necessary building blocks to build real scripts.

Chapter 3, "PHP 5 OO Language," continues exploring PHP 5's syntax, focusing on its object-oriented functionality. This chapter covers basics, such as properties and methods, and progresses to more complicated subjects, such as polymorphism, interfaces, exceptions, and lots more.

Using the previous chapter as a foundation, Chapter 4, "PHP 5 Advanced OOP and Design Patterns," covers some of the most advanced features of PHP 5's object model. After learning these features, including four commonly used design patterns and PHP's reflection capabilities, you will soon become an OO wizard.

Now that you are familiar with the syntax and language features of PHP, Chapter 5, "How to Write a Web Application with PHP," introduces you to the world of writing web applications. The authors show you basics, such as handling input through form variables and safety techniques, but this chapter also includes more advanced topics, such as handling sessions with cookies and PHP's session extension. You also find a few tips on laying out your source code for your web applications.

Chapter 6, "Databases with PHP 5," introduces using MySQL, SQLite, and Oracle from PHP, but focuses primarily on the PHP 5-specific details of database access. For each database, you learn about some of its strong and weak points, as well as the types of applications at which each excels. And of course, you learn how to interface with them using PHP's native functions or using PEAR DB.

All scripts can throw errors, but of course you do not want them to show up on your web site once your application has passed its development state. Chapter 7, "Error Handling," deals with different types of errors that exist, how to handle those errors with PHP, and how to handle errors with PEAR.

As one of the important new features in PHP 5 is its renewed XML support, a chapter on XML features in PHP 5 could not be missed. Chapter 8, "XML with PHP 5," talks about the different strategies of parsing XML and converting XML to other formats with XSLT. XML-RPC and SOAP are introduced to show you how to implement web services with both techniques.

Although not specifically for PHP 5, the five mainstream extensions that Chapter 9,"Mainstream Extensions," covers are important enough to deserve a place in this book. The first section, "Files and Streams," explains about handling files and network streams. A **stream** is nothing more than a way to access external data, such as a file, remote URL, or compressed file. The second section, "Regular Expressions," explains the syntax of a regular expression engine (PCRE) that PHP uses with numerous examples to show you how these expressions can make your life easier. In "Date Handling," we explain the different functions used to parse and format date and time strings. In "Graphics Manipulation with GD," we show you through two real-life scenarios the basic functions of creating and manipulating graphics with PHP. The last section in this chapter, "Multibyte Strings and Character Sets," explains the different character sets and the functions to convert and handle different ones, including multi-byte strings used in Asian languages.

Chapter 10, "Using PEAR," introduces PEAR, the PHP Extension and Application Repository. Starting with concepts and installation, the chapter shows how to use PEAR and maintain the local installed packages. This chapter also includes a tour of the PEAR web site.

Chapter 11, "Important PEAR Packages," gives an overview of the most important PEAR packages, along with examples. Packages covered include Template Systems, the **Auth** package to do authentication, form handling with the `HTML_QuickForm` package, and a package used to simplify caching.

Chapter 12, "Building PEAR Components," explains how to create your own PEAR package. The PEAR Coding Standard and `package.xml` package definition format, together with tips on including files and package layout, get you on your way to completing your first PEAR package.

Chapter 13, "Making the Move," deals with the few backward-incompatible changes that were introduced between PHP 4 and PHP 5. This chapter tells you which things you need to take care of when making your application work on PHP 5, and provides workarounds wherever possible.

Chapter 14, "Performance," shows you how to make your scripts perform better. The chapter offers tips on standard PHP usage, the use of external utilities (APD and Xdebug) to find problems in your scripts, and PHP accelerators like APC and Zend Performance Suite.

Chapter 15, “An Introduction to Writing PHP Extensions,” explains how to write your own custom PHP extension. We use a simple example to explain the most important things like parameter parsing and resource management.

Chapter 16, “PHP Shell Scripting,” shows you how to write shell scripts in PHP, because PHP is useful for more than just web applications. We carefully explain the differences between the CLI and CGI executables in which PHP comes, including command-line parameter parsing and process control.

This book also includes three appendices. Appendix A, “PEAR and PECL Package Index,” provides an overview of all important packages, with descriptions and dependencies on other packages. Appendix B, “phpDocument Format Reference,” explains the syntax as understood by the PHP Documenter tool to generate API documentation from source code. Appendix C, “Zend Studio Quick Start,” is an introduction to working in the Zend Studio IDE.

A Note About Coding Styles

There are almost as many coding styles as there are programmers. The PHP examples in this book follow the PEAR coding standard, with the opening curly bracket on the line below the function name. In some cases, we’ve placed the curly bracket on the same line as the function name. We encourage you to adopt the style you are most comfortable with.

Note: A code continuation character, ↪, appears at the beginning of code lines that have wrapped down from the line above it.

About the Software

Included in the back of this book is a special link to Zend.com, where you can download a fully functional, 90-day trial version of the Zend Studio IDE. Be sure to use the license key printed on the inside back cover of this book when you install Zend Studio.

The Zend Development Environment (ZDE) is a convenient tool that integrates an editor, debugger, and project manager to help you develop, manage, and debug your code. It can connect to your own installed server or directly to the Zend Studio server component. It is a powerful tool that allows you to debug your code in its natural environment.

Updates and Errata and Downloads

Updates, errata, and copies of the sample programs used in this book can be found at the following URL: http//php5powerprogramming.com. We encourage you to visit this site.

ACKNOWLEDGEMENTS

This book could not have been written without feedback from our technical reviewers; therefore, we would like to thank Marcus Börger, Steph Fox, Martin Jansen, and Rob Richards for their excellent comments and feedback. Besides these four reviewers, there are a few more people who helped answer several questions during the writing of this book, more specifically Christian Stocker for helping with the XML chapter, Wez Furlong and Sara Golemon for answering questions about the streams layer, Pierre-Alain Joye for providing some insights in the inner workings of the GD library, and less specifically the PEAR community for their support and dedication to a great repository of usable PEAR components. Some sections in this book were contributed by co-authors; Georg Richter contributed the MySQLi section of the database chapter, and Zeev Suraski added the section on Zend's Performance Suite.

We would also like to thank Mark L. Taub and the editorial team of Pearson PTR for the things they are good at doing: organizing, planning, and marketing this book, and making sure everything fits together. Thanks to Janet Valade, for helpful developmental editing support, and our project editor Kristy Hart, who helped us wrap up the book under pressure and put the final touches on it.

Enjoy!
Andi, Stig, and Derick