Unlock PHP 8: From Basic to Advanced

The next-level PHP 8 guide for dynamic web development

Roni Sommerfeld

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55519-757

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My wife Isabel and my daughter, ***Anne****,*

Who are the true architects of my heart, who taught me that love is the perfect algorithm: resilient, adaptable, and infinitely surprising.

About the Author

Roni Sommerfeld, Brazilian and German, is a technology enthusiast with extensive experience in system development since 2010. He has worked on highly complex projects, handling over 50 million records daily with PHP. As the founder of the PHPiando blog and YouTube channel, he shares tips and insights on programming. Living in Portugal with his wife Isabel, daughter Anne, and their dog Odin, he swaps code for cooking, blending culinary recipes with programming ones in a delightful harmony.

About the Reviewer

Marco Dal Monte has a Master's Degree in Computer Science from the University of Padova, Italy, where he specialized his studies in Artificial Intelligence. After university, he worked for a brief period in Italy before moving to London.

He works mainly in PHP, using Symfony and Laravel as frameworks, and he is interested in software engineering methodologies, DevOps and system administration in Linux. Currently, he is working at RemitONE, a company based in London that is a leader in white label software for money transfers.

Acknowledgement

The journey to completing this book has been marked by support and inspiration from many valuable sources.

I am immensely grateful to my wife, Isabel, and our daughter, Anne, whose love and support were fundamental. You, along with the rest of our family, are the true support and joy in my life. Thank you for being by my side every step of this process.

Special thanks to the BPB team, whose expertise and dedication turned a vision into reality. The collaboration of reviewers, technical experts, and editors was essential in improving each chapter of this book.

I extend my gratitude to my professional colleagues and friends who shared their experiences and knowledge with me. You have enriched my professional and personal perspective, contributing to the content and depth of this work.

And last but not least, I thank you, the reader, for embarking on this journey with me. Their quest for knowledge and growth is what motivates authors to create and share their works. I hope this book offers you valuable insights and inspiration for your projects.

Preface

The constant evolution of technology requires developers to stay up to date with the latest innovations, practices, and programming languages. PHP 8, with its new features and significant improvements, marks an important step in this ongoing journey of learning and development.

This book is designed as a comprehensive guide to exploring PHP 8 in depth, from the basic fundamentals to the advanced features that define this version. Covering topics such as functions, form management, sessions and cookies, arrays and collections, **object-oriented programming** (**OOP**), error and exception handling, data persistence, and advanced development practices, the book is an indispensable resource for anyone involved in web development.

Throughout this book, you will discover the key features of PHP 8 and learn how to apply them to create efficient, reliable, and maintainable web applications. Security and performance optimization best practices will also be discussed, providing you with a solid foundation for designing and implementing robust applications.

This book is aimed at developers who are starting out with PHP and want to learn how to build web applications, as well as experienced developers looking to improve their knowledge in this new version. With practical examples and a step-by-step approach, you will be guided through the essential concepts to become proficient in using PHP 8.

With the support of this book, you will acquire the knowledge and skills necessary to excel in developing modern web applications, making the most of the innovations that PHP 8 has to offer. I hope you find this book a valuable source of information and inspiration.

Chapter 1 - Introduction to PHP 8: This chapter introduces PHP, covering its origins and the evolution of PHP 8. It guides readers through setting up their development environment with Laragon and LAMP, offers tips for migrating to PHP 8, and discusses the future prospects of PHP, providing a comprehensive overview for starting PHP development.

Chapter 2 - Fundamentals with PHP 8: The chapter explores the core principles of PHP programming, including variable types, control structures, and basic data manipulation. The chapter lays a solid foundation for understanding how PHP processes information and how to write simple, effective code.

Chapter 3 - Functions in PHP: This chapter dives into the creation and usage of functions in PHP, highlighting built-in functions as well as how to define custom functions. It covers

function syntax, passing arguments, return values, and the importance of functions in modular coding.

Chapter 4 - Forms, Sessions and Cookies: This chapter discusses how to handle user input through forms, maintain session states, and use cookies to personalize user experiences. This chapter provides practical examples to secure data handling and user authentication.

Chapter 5 - Arrays and Collections: This chapter focuses on managing groups of data using arrays and collection tools in PHP. It explains array operations, associative arrays, and the use of built-in functions to sort, filter, and manipulate data collections.

Chapter 6 - OOP Advanced Features of PHP 8+: The chapter introduces **object-oriented programming** (**OOP**) concepts in PHP 8+, including classes, objects, inheritance, and polymorphism. This chapter emphasizes the new OOP features and best practices for designing reusable and maintainable code.

Chapter 7 - Handling Errors and Exceptions: This chapter covers strategies for robust error handling and exception management in PHP applications. It guides readers through PHP's error reporting mechanisms, custom exception handling, and tips for debugging code efficiently.

Chapter 8 - Database and Data Persistence with PHP: This chapter explores how to interact with databases using PHP, with a focus on PDO and MySQLi extensions. The chapter provides insights into performing CRUD operations, preventing SQL injection, and managing data persistence.

Chapter 9 - Advanced Development with PHP: We will cover the advanced capabilities of PHP in its recent versions. Exploring the operation and application of RESTful APIs, especially with regard to communication with external services. We will demonstrate best practices for establishing APIs in PHP and the language's most recent additions: Fibers and enums.

Chapter 10 - Best Practices Security and Performance with PHP: This chapter concludes the book by addressing critical aspects of web development: security and performance optimization. It discusses secure coding practices, data encryption, safeguarding against common vulnerabilities, and techniques to enhance the speed and efficiency of PHP applications.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/em31l9g

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Unlock-PHP-8-From-Basic-to-Advanced.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Introduction to PHP 8

Introduction

In this first chapter, we will do a little introduction and analysis of PHP versions and explore the evolution of the language from version 5 to the most recent update, PHP 8+. In its evolution, PHP has increased and is continuously introducing new features, thus becoming one of the outstanding programming languages in terms of web development.

Structure

The chapter covers the following topics:

- The origin of PHP
- Evolution of major PHP versions
- Creating development environment
- Tips on migrating to PHP 8
- Considerations and the future of PHP

Objectives

After reading this chapter, you will be able to address the main changes and improvements introduced in these new versions. In addition to addressing the migration of projects

developed in PHP 5 to PHP 8+, we will also highlight new features and improvements that make PHP 8+ a good choice for developers. Finally, you will be able to understand PHP's future trends and expectations, giving an overview of what developers can expect from this popular programming language.

The origin of PHP

Before we get into any kind of content in this book, it is important for us to remember the origin and journey of PHP, one of the most widely used server-side scripting languages, started in 1994. Originally created by *Rasmus Lerdorf*, a programmer from Greenland, PHP started as a simple set of CGI scripts for tracking visitors to your online resume. The original name of PHP was **Personal Home Page Tools**, a primitive but functional way of building dynamic websites.

Realizing the potential of what he had created, *Lerdorf* decided to improve his scripts, adding features for working with web forms and communicating with databases. In 1995, he released the PHP/FI (*Form Interpreter*) version, which formed the basis of what would become the PHP we know today.

In version PHP 3, the language started to look more like the modern form of PHP, thanks to the work of *Andi Gutmans* and *Zeev Suraski*. They needed a more dynamic language for the project they were developing at the time at the university, so they rewrote and expanded the PHP language, making it a more robust and flexible solution for web development.

PHP 4, released in 2000 as the work of *Andi* and *Zeev*, brought several improvements, including more object-oriented features. However, it was PHP 5, released in 2004, that solidified PHP's position as one of the top programming languages for web development. PHP 5 introduced a complete object model, as well as error handling improvements and several powerful extensions.

Throughout its evolution, PHP has managed to maintain a careful balance between ease of use and power, which has contributed to its enduring popularity as a web programming language. An example of a PHP code is illustrated in the following figure:

```
<!--include /text/header.html-->

<!--getenv HTTP_USER_AGENT-->
<!--ifsubstr $exec_result Mozilla-->
  Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='$username'-->
<!--ifless $numentries 1-->
  Sorry, that record does not exist<p>
<!--endif exit-->
  Welcome <!--$user-->!<p>
  You have <!--$index:0--> credits left in your account.<p>

<!--include /text/footer.html-->
```

Figure 1.1: *PHP/FI code*[1]

[1] source: **https://www.php.net/manual/en/history.php.php**

Evolution of major PHP versions

Let us learn a bit more about the PHP releases that have had the greatest strengths.

PHP5

The release of PHP 5 in July 2004 was a major milestone in PHP history. This release included additional new features and improvements related to PHP 4 as it had a significant impact on websites where web development was not done with PHP.

One of the main features of PHP 5 is the introduction of a full object model. Before PHP 5, the language had specific functions for programming a specific object (OOP), but only to a limited extent. Classes and objects are more powerful and flexible in PHP 5. Classes can give the programmatic object a more beautiful and practical look and better effectiveness due to restricted visibility (public, private, and protected), and due to its properties and methods (class functions).

PHP 5 also introduces exceptions inspired by other languages like Java and C++. Exceptions are used to make errors more effective and to help developers constantly detect and then check for errors and checks during script execution.

Another benefit of PHP 5 is the introduction of new and more powerful extensions. For example, the PDO and MySQLi extensions provide new functionality for connecting to MySQL databases. The SimpleXML extension simplifies the editing and analysis of XML data. It also added SOAP pricing to ease integration with web-based services over SOAP.

PHP 5 is the first programming language for the web, giving developers a greater level of control and flexibility, as well as ease of use. This enables the creation of complex and more powerful web applications as well as the implementation of vague innovation projects on the web.

PHP 5.6

PHP 5.6, released in August 2014, was the last major release of PHP 5 and brought several significant new features and improvements. The main goal of PHP 5.6 was to increase language consistency, improve performance, and add functionality requested by users and the development community.

Among the most important and striking features of PHP 5.6, we can highlight the following:

- **Variable argument support:** PHP 5.6 introduced the splat operator syntax (`...`) for representing variable arguments in functions and methods. This allowed developers to pass a variable number of arguments to a function, making PHP more flexible and less error prone.

  The following is an example of using the splat operator (...). We are defining a function that can take any number of arguments and compute the sum of all of them:

```php
.<?php
function sumNumbers(...$numbers) {
 $total = 0;
 foreach ($numbers as $number) {
   $total += $number;
 }
 return $total;
}

echo sumNumbers(1, 2, 3, 4, 5);  // Output: 15
```

- **Importing functions and namespace constants:** Importing functions and namespace constants allowed users to simplify the code and increase readability.

  In the following example, we see how the code can be simplified and made more readable by importing functions and namespace constants. We define a function and a constant in a namespace and then import them to use:

```php
<?php
namespace MyNamespace;

function myFunction() {
 echo "It's works";
}

const MY_FIRST_CONST = 123;
```

For example:

```php
<?php
use MyNamespace\myFunction;
use MyNamespace\MY_FIRST_CONST;


myFunction();  // Output: It's works
echo MY_FIRST_CONST;  // Output: 123
```

- **Exponentiation operator:** PHP 5.6 introduced the exponentiation operator (**), making it easier to perform complex mathematical operations.

Here is a simple example of how the exponentiation operation works. In this case, we are raising 2 to the power of 3:

```
<?php
$total = 2 ** 3;  // 2 raised to the power of 3
echo $total;  // Output: 8
```

- **GMP compatibility in common functions:** Arbitrary precision number operations became easier with PHP 5.6 due to compatibility with the **GNU Multiple Precision (GMP)** library.

  This example demonstrates the ease of performing operations with arbitrary precision numbers with the GMP library. We are adding two numbers of arbitrary precision:

```
<?php
$number1 = gmp_init(123);
$number2 = gmp_init(456);

$total = gmp_add($number1, $number2);
echo gmp_strval($total);  // Output: 579
```

- **Hash function improvements:** PHP 5.6 introduced **`hash_equals()`** for secure string comparison, which significantly improved the security of string handling in PHP.

  The example below shows how to use the **`hash_equals()`** function for safe string comparison. In this case, **`'hello'`** and **`'HELLO'`** are different, so the result will be **`'Strings are different.'`**:

```
<?php
$string1 = 'hello';
$string2 = 'HELLO';

if (hash_equals($string1, $string2)) {
 echo 'The strings are the same.';
} else {
 echo 'Strings are different.';
}
```

- **Deeper integration of Zend OPcache:** OPcache, which compiles PHP code into bytecode to speed up its performance, has been further integrated in PHP 5.6, significantly improving the performance of PHP applications.