PHP & Laravel Interview Master Guide — 150+ Questions with Answers

Written by Neeraj Saini — Senior Software Engineer

aka HaxNeeraj www.haxneeraj.com

Preface

Welcome to the **PHP & Laravel Interview Master Guide**, your complete resource to prepare for PHP and Laravel interviews — from beginner to advanced levels.

This eBook is designed to help developers strengthen their fundamentals, understand real-world use cases, and answer questions confidently during interviews.

Each question includes a **clear, short explanation** and **practical example** based on PHP 8.3 and Laravel 11 standards.

Let's begin your journey to becoming a confident and job-ready PHP/Laravel developer.

Section 1: Core PHP Interview Questions (Q1–Q30)

Q1. What is PHP and why is it popular?

Answer: PHP (Hypertext Preprocessor) is a server-side scripting language used for web development. It's popular because:

- It's open-source and widely supported.
- It integrates easily with HTML, MySQL, and Apache.
- It powers frameworks like Laravel, Symfony, and WordPress.

Example:

```php
<?php
echo "Hello, World!";
?>
```

Q2. What are the key features of PHP 8+?

Answer: PHP 8 introduced several performance and syntax improvements:

- **JIT Compiler** for faster execution.

- **Union Types** for better type hinting.
- **Attributes (Annotations)** for metadata.
- **Match Expression** as a cleaner alternative to `switch`.
- **Named Arguments** and **Constructor Property Promotion**.

Q3. What is the difference between echo, print, and print_r()?

Answer:

Function	Usage	Returns	Example
`echo`	Outputs one or more strings	No	`echo "Hello";`
`print`	Outputs a single string	Returns 1	`print("Hi");`
`print_r()`	Prints human-readable info about arrays/objects	No	`print_r($array);`

Q4. What is the difference between == and === in PHP?

Answer:

- `==` → Compares values (type conversion allowed).
- `===` → Compares both value and data type.

Example:

```
0 == "0"    // true
0 === "0"   // false
```

Q5. What are variables and constants in PHP?

Answer:

- **Variable** → Stores data, defined with `$`, mutable.
- **Constant** → Defined using `define()` or `const`, immutable.

Example:

```
$name = "Neeraj";
define('VERSION', '1.0');
```

Q6. What are data types in PHP?

Answer: PHP supports 8 primitive data types:

- Scalar: `int`, `float`, `string`, `bool`

- Compound: `array`, `object`
- Special: `null`, `resource`

Q7. What is type juggling in PHP?

Answer: Type juggling means PHP automatically converts data types during operations. Example:

```
echo 5 + "5"; // 10 (string converted to int)
```

Q8. What is type hinting?

Answer: Type hinting enforces a variable's data type for function parameters or return values.

Example:

```
function sum(int $a, int $b): int {
  return $a + $b;
}
```

Q9. What are superglobals in PHP?

Answer: Built-in variables accessible from anywhere: `$_GET`, `$_POST`, `$_SERVER`, `$_COOKIE`, `$_SESSION`, `$_FILES`, `$_ENV`, `$_REQUEST`.

Q10. Difference between GET and POST methods

Method	Data Visibility	Security	Data Limit
GET	Visible in URL	Less secure	~2000 chars
POST	Hidden in body	More secure	No limit

Q11. What are Sessions and Cookies?

Answer:

- **Session:** Server-side, expires when browser closes.
- **Cookie:** Client-side, stored in browser.

Example:

```
// Session
session_start();
$_SESSION['user'] = 'Neeraj';

// Cookie
setcookie('user', 'Neeraj', time()+3600);
```

Q12. What is the difference between include, require, and include_once?

Function	Description
`include`	Gives a *warning* if file missing
`require`	Gives a *fatal error* if file missing
`include_once`	Prevents multiple inclusions

Q13. What is an array in PHP?

Answer: A data structure to store multiple values in one variable.

Example:

```
$colors = ['red', 'green', 'blue'];
echo $colors[1]; // green
```

Q14. What are associative and multidimensional arrays?

Example:

```
// Associative
$user = ['name' => 'Neeraj', 'role' => 'Engineer'];

// Multidimensional
$users = [
  ['name' => 'Neeraj', 'role' => 'Engineer'],
  ['name' => 'Subham', 'role' => 'HR']
];
```

Q15. What are control structures in PHP?

Answer: Conditional and looping constructs like: `if`, `else`, `elseif`, `switch`, `for`, `foreach`, `while`, `do...while`.

Q16. What are functions in PHP?

Answer: Reusable blocks of code.

```
function greet($name) {
  return "Hello, $name!";
}
echo greet('Neeraj');
```

Q17. What is variable scope in PHP?

Answer:

- Local → within function
- Global → accessible everywhere using `global` keyword
- Static → retains value between calls

Q18. What is recursion in PHP?

Answer: Function calling itself until a condition is met.

```
function factorial($n) {
  return $n <= 1 ? 1 : $n * factorial($n - 1);
}
```

Q19. What is difference between `unset()` and `unlink()`?

- `unset()` → Deletes a variable.
- `unlink()` → Deletes a file.

Q20. What are magic methods in PHP?

Answer: Predefined methods starting with `__`: `__construct()`, `__destruct()`, `__get()`, `__set()`, `__toString()`, etc.

Q21. What is object-oriented programming (OOP)?

Answer: OOP organizes code into classes and objects to promote reusability and scalability.

Q22. What is the difference between class and object?

Term	Description
Class	Blueprint for objects
Object	Instance of a class

Q23. What is inheritance in PHP?

Answer: Child class inherits parent class properties and methods.

```
class A { public function greet() { echo "Hi"; } }
class B extends A {}
(new B)->greet();
```

Q24. What is polymorphism?

Answer: Same method name but different implementation across classes.

Q25. What are interfaces and abstract classes?

Answer:

- **Interface:** Only defines method signatures.
- **Abstract:** Can have both abstract and normal methods.

Q26. What are traits in PHP?

Answer: Traits allow method reuse across multiple classes.

```
trait Logger { public function log($msg) { echo $msg; } }
class App { use Logger; }
```

Q27. What is exception handling in PHP?

Answer: Using `try`, `catch`, and `finally`.

```
try {
  throw new Exception("Error occurred");
} catch (Exception $e) {
  echo $e->getMessage();
}
```

Q28. What is PDO in PHP?

Answer: PDO (PHP Data Objects) provides a secure, OOP way to interact with databases.

Q29. What is prepared statement?

Answer: Precompiled SQL queries preventing SQL injection.

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE email = ?");
$stmt->execute([$email]);
```

Q30. What are design patterns in PHP?

Answer: Reusable solutions to common problems:

- Singleton
- Factory
- Repository
- Strategy
- Observer

End of Part 1 (Core PHP)

Visit this link to access all parts

[https://github.com/haxneeraj/php-laravel-interview-questions-answers-part-1%5D(https://github.com/haxneeraj/php-laravel-interview-questions-answers-part-1)