67 PHP interview questions to ask your candidates

Questions

1. Can you explain the difference between `echo` and `print` in PHP?
2. What are the different types of errors in PHP, and how do you handle them?
3. How do you connect to a MySQL database using PHP?
4. What are some common functions used for string manipulation in PHP?
5. Describe how you would handle form data in PHP.
6. Can you explain the concept of sessions and cookies in PHP?
7. What is the purpose of the `composer.json` file in a PHP project?
8. How do you implement error handling in PHP?
9. Describe the Model-View-Controller (MVC) architecture and how it is implemented in PHP.
10. What are traits in PHP, and how are they used?
11. What are some key features of PHP?
12. How do you secure a PHP application?
13. Can you explain the purpose and use of PHP frameworks?
14. What are some common methods for debugging PHP code?
15. How do you manage dependencies in a PHP project?
16. What is the difference between include() and require() in PHP?
17. How do you handle file uploads in PHP?
18. Can you explain the concept of namespaces in PHP?
19. What are the advantages of using Object-Oriented Programming (OOP) in PHP?
20. Can you explain how PHP's garbage collection mechanism works?
21. How do you implement pagination in PHP?
22. What are PSR standards, and why are they important in PHP development?
23. Can you describe how to use prepared statements in PHP and why they are beneficial?
24. What is the difference between public, private, and protected visibility in PHP classes?
25. How does PHP handle type juggling, and what are some potential pitfalls?
26. Explain how to use cURL in PHP to make HTTP requests.
27. How do you implement caching in a PHP application?
28. Can you describe the difference between GET and POST methods in form submission and when to use each?
29. What are anonymous functions (closures) in PHP and how are they used?
30. How would you handle large file uploads in PHP?
31. Can you explain how to use the PHP Data Objects (PDO) extension for database interactions?
32. What are the benefits and drawbacks of using PHP for web development?
33. How do you handle and validate user input in PHP?
34. What are some common security risks in PHP applications, and how do you mitigate them?
35. Can you explain the difference between static and dynamic methods in PHP?
36. How do you implement and use interfaces in PHP?
37. Can you describe the SPL (Standard PHP Library) and its significance?
38. Can you explain what PHP stands for and its primary purpose?
39. What is the difference between single-quoted and double-quoted strings in PHP?
40. Can you explain what a PHP session is and how it works?
41. What is the purpose of the 'finally' block in exception handling?
42. Can you explain what a PHP trait is and when you might use one?
43. What is the difference between `isset()` and `empty()` in PHP?
44. Can you explain what a PHP namespace is and why it's useful?
45. What is the purpose of the `__construct()` method in PHP classes?
46. How would you manage a long-running process in PHP without causing a timeout?
47. Can you explain how to run a background task in PHP?
48. Describe the method you would use to execute a shell command from within a PHP script.
49. What are some effective ways to optimize a PHP script's performance?
50. How do you handle concurrent processes in PHP?
51. Can you explain how to use cron jobs with PHP?
52. Describe how you would monitor and log PHP processes for debugging and performance analysis.
53. How do you implement task scheduling in a PHP application?
54. What strategies would you use to handle asynchronous tasks in PHP?
55. How can you ensure that a PHP script can handle multiple tasks simultaneously?
56. Can you describe a method for executing parallel tasks in PHP?
57. How would you manage resource-intensive tasks within a PHP application?
58. How would you approach debugging a PHP application that is running very slowly?
59. If a client reports that a feature is not working as expected, how would you go about troubleshooting and resolving the issue?
60. Describe a time when you had to optimize a PHP script. What steps did you take and what was the outcome?
61. Imagine you're tasked with migrating an application from one server to another. What considerations would you keep in mind during this process?
62. How would you handle a situation where you need to integrate a third-party API into your existing PHP application?
63. What steps would you take if you need to add new functionality to a legacy PHP application without breaking existing features?
64. If you discover a security vulnerability in your PHP application, what immediate actions would you take to address it?
65. Describe how you would implement a feature that requires real-time data updating in a PHP application.
66. If a team member disagrees with your proposed solution to a problem, how would you handle that situation?