Top 50 PHP Interview Questions in 2024

Topics : PHP Interview Questions
Written on November 18, 2023

Basic PHP Concepts:

1. **What is PHP, and how does it differ from other scripting languages?**

   Answer: PHP (Hypertext Preprocessor) is a server-side scripting language designed for web development. It differs from other scripting languages by embedding directly into HTML code.

2. **Explain the difference between == and === in PHP.**

   Answer: == checks for equality after type coercion, while === checks for equality without type coercion.

3. **What is the purpose of the `echo` statement in PHP?**

   Answer: `echo` is used to output one or more strings or variables.

4. **How can you include a file in PHP?**

   Answer: Files can be included using the `include` or `require` statements.

5. **Explain the use of the `isset()` function in PHP.**

   Answer: `isset()` is used to check if a variable is set and not `null`.

PHP Syntax and Language Features:

6. **How do you declare a constant in PHP?**

   Answer: Constants are declared using the `define()` function.

7. **Explain the purpose of the `foreach` loop in PHP.**

   Answer: `foreach` is used to loop through each key/value pair in an array.

8. **What is the ternary operator, and how is it used in PHP?**

   Answer: The ternary operator (`? :`) is a shorthand way to write an `if-else` statement.

9. **What is the purpose of the `empty()` function in PHP?**

Answer: `empty()` is used to check if a variable is empty.

10. **Explain the difference between `include()` and `require()` in PHP.**

    Answer: Both include and require are used to include files, but `require()` will produce a fatal error if the file is not found, while `include()` only produces a warning.

Web Development with PHP:

11. **How can you pass data between the client and server in PHP?**

    Answer: Data can be passed through forms using `GET` or `POST` methods, and sessions or cookies can be used for persistent data.

12. **What is the difference between `GET` and `POST` methods in PHP?**

    Answer: `GET` appends data to the URL, while `POST` sends data in the HTTP request body.

13. **Explain the purpose of the `$_SESSION` superglobal in PHP.**

    Answer: `$_SESSION` is used to store session variables that persist across multiple pages.

14. **How can you set and retrieve cookies in PHP?**

    Answer: Cookies can be set using `setcookie()` and retrieved using `$_COOKIE`.

15. **What is URL rewriting in PHP, and how is it implemented?**

    Answer: URL rewriting is the process of altering or rewriting a URL to achieve specific functionalities. It is implemented using `.htaccess` file for Apache or server configuration.

PHP Functions:

16. **How do you declare a function in PHP?**

    Answer: Functions are declared using the `function` keyword.

17. **Explain the difference between `return` and `echo` in PHP functions.**

    Answer: `return` is used to return a value from a function, while `echo` is used to output data.

18. **What is variable scope in PHP?**

    Answer: Variable scope defines where a variable can be accessed. PHP has local, global, and static scope.

19. **How can you pass parameters to a PHP function?**

    Answer: Parameters can be passed directly in the function declaration.

20. **Explain the use of the `static` keyword in PHP functions.**

    Answer: The `static` keyword is used to declare a static method or property, which belongs to

the class rather than an instance of the class.

Arrays and Data Structures:

21. **How do you create an array in PHP?**

    Answer: Arrays can be created using the `array()` constructor or shorthand `[]` syntax.

22. **Explain the concept of associative arrays in PHP.**

    Answer: Associative arrays use named keys rather than numerical indices.

23. **What is the purpose of the `array_merge()` function in PHP?**

    Answer: `array_merge()` is used to merge two or more arrays.

24. **How can you sort an array in PHP?**

    Answer: Arrays can be sorted using functions like `sort()`, `asort()`, `ksort()`, etc.

25. **Explain the concept of multidimensional arrays in PHP.**

    Answer: Multidimensional arrays are arrays within arrays, forming a matrix-like structure.

PHP and Databases:

26. **How can you connect to a MySQL database in PHP?**

    Answer: MySQL connection is established using the `mysqli` or `PDO` extension.

27. **What is SQL injection, and how can it be prevented in PHP?**

    Answer: SQL injection is a security vulnerability. It can be prevented by using prepared statements and parameterized queries.

28. **How can you retrieve data from a MySQL database in PHP?**

    Answer: Data can be retrieved using SQL queries and functions like `mysqli_query()`.

29. **What is the purpose of the `mysqli_fetch_assoc()` function in PHP?**

    Answer: `mysqli_fetch_assoc()` is used to fetch a result row as an associative array.

30. **How can you update data in a MySQL database using PHP?**

    Answer: Data can be updated using SQL `UPDATE` queries and functions like `mysqli_query()`.

Error Handling and Debugging:

31. **What is the purpose of the `error_reporting` directive in PHP?**

Answer: It controls the level of error reporting.

32. **Explain the use of `try`, `catch`, and `finally` blocks in PHP for exception handling.**

    Answer: They are used for catching and handling exceptions. `finally` block is optional.

33. **What is the purpose of the `die()` function in PHP?**

    Answer: `die()` is used to output a message and terminate the script.

34. **Explain the use of the `var_dump()` function in PHP.**

    Answer: `var_dump()` is used to display structured information (type and value) about variables.

35. **How can you enable or disable error display in PHP?**

    Answer: Error display can be controlled using the `display_errors` directive in the `php.ini` file.

Object-Oriented Programming (OOP) in PHP:

36. **What is OOP, and how is it implemented in PHP?**

    Answer: OOP is a programming paradigm that uses objects and classes. PHP supports OOP with classes and objects.

37. **Explain the concepts of encapsulation, inheritance, and polymorphism in PHP.**

    Answer: Encapsulation is the bundling of data and methods that operate on the data. Inheritance is the ability of a class to inherit properties and methods from another class. Polymorphism allows objects of different types to be treated as objects of a common type.

38. **How do you declare a class in PHP?**

    Answer: Classes are declared using the `class` keyword.

39. **Explain the use of the `public`, `private`, and `protected` keywords in PHP classes.**

    Answer: They control the visibility of class properties and methods.

40. **What is the purpose of the `__construct()` method in PHP classes?**

    Answer: `__construct()` is a constructor method that is automatically called when an object is created.

PHP Security:

41. **What is Cross-Site Scripting (XSS), and how can it be prevented in PHP?**

    Answer: XSS is a security vulnerability. It can be prevented by validating and sanitizing user input and using secure coding practices.

42. **Explain the concept of Cross-Site Request Forgery (CSRF) and its prevention in PHP.**

    Answer: CSRF is an attack that forces an end user to perform undesired actions on a web application. Prevention involves using anti-CSRF tokens.

43. **What is a session hijacking attack, and how can it be prevented in PHP?**

    Answer: Session hijacking involves stealing session information. Prevention includes using secure connections (HTTPS) and session timeout settings.

44. **Explain the concept of prepared statements and how they help prevent SQL injection.**

    Answer: Prepared statements are precompiled SQL statements. They prevent SQL injection by separating SQL code from user input.

45. **How can you sanitize user input in PHP?**

    Answer: User input can be sanitized using functions like `filter_var()` and `htmlspecialchars()`.

Web Services and APIs in PHP:

46. **What is an API, and how can you consume an API in PHP?**

    Answer: An API (Application Programming Interface) allows communication between different software systems. It can be consumed using functions like `file_get_contents()` or cURL.

47. **Explain the use of cURL in PHP for making HTTP requests.**

    Answer: cURL is a library for making HTTP requests. It can be used in PHP to send and receive data over HTTP.

48. **What is RESTful API, and how can you create one using PHP?**

    Answer: A RESTful API is an architectural style for designing networked applications. It can be created in PHP using frameworks like Laravel or by manually handling HTTP requests.

49. **Explain the purpose of the `json_encode()` and `json_decode()` functions in PHP.**

    Answer: `json_encode()` is used to convert a PHP object or array to a JSON string, and `json_decode()` is used to convert a JSON string to a PHP object or array.

50. **How can you handle authentication in a PHP-based API?**

    Answer: Authentication can be handled using tokens, API keys, or OAuth.