

PHP Interview Questions

PHP: Theory Interview Questions

1. PHP Basics

- **Q1:** What is PHP, and how does it differ from other server-side scripting languages?
- **Q2:** Explain the difference between GET and POST methods in PHP.
- **Q3:** What are sessions and cookies in PHP? How are they used?
- **Q4:** What is the difference between `include()`, `require()`, `include_once()`, and `require_once()` in PHP?
- **Q5:** How does error handling work in PHP?

2. PHP Data Handling

- **Q1:** Explain the difference between `$_GET` and `$_POST` in PHP.
- **Q2:** What is a superglobal in PHP, and which superglobals are commonly used?
- **Q3:** How can you sanitize and validate user input in PHP?
- **Q4:** What is the purpose of the `isset()`, `empty()`, and `unset()` functions?
- **Q5:** What is the use of the `file_get_contents()` function?

3. PHP Functions and OOP

- **Q1:** Explain the difference between passing by value and passing by reference in PHP.
- **Q2:** What are anonymous functions (closures) in PHP?
- **Q3:** Explain the difference between static and dynamic methods in PHP.
- **Q4:** What are namespaces in PHP, and why are they useful?
- **Q5:** What are interfaces and abstract classes in PHP? How are they used?

4. PHP Database Integration

- **Q1:** What are the steps to connect to a MySQL database in PHP?
- **Q2:** What are prepared statements in PHP, and why are they important?
- **Q3:** How does PHP handle SQL injection attacks, and how can it be prevented?
- **Q4:** Explain the use of PDO in PHP. What advantages does it offer over `mysqli`?
- **Q5:** What is the difference between `mysqli_fetch_assoc()` and `mysqli_fetch_array()`?

PHP: Practical Interview Questions

1. PHP Basics

- **Q1:** Write a PHP script that prints the current date and time.
- **Q2:** Write a PHP script to create and store cookies and retrieve their values.
- **Q3:** Create a simple form that submits data using POST. Process the form data on the server side and display it.

- **Q4:** Write a program to display all prime numbers between 1 and 100.
- **Q5:** Write a PHP script to check if a string is a palindrome.

2. PHP Data Handling

- **Q1:** Create a PHP script that sanitizes and validates an email input.
- **Q2:** Write a PHP program to read and write to a file.
- **Q3:** Write a PHP script that uses `$_GET` to retrieve parameters from a URL and display them.
- **Q4:** Write a script that counts the number of words in a user-inputted string.
- **Q5:** Create a PHP script that reads a CSV file and outputs its contents in a table format.

3. PHP OOP

- **Q1:** Write a PHP class for a `Car` that includes properties like `make`, `model`, and methods for `startEngine` and `stopEngine`.
- **Q2:** Create a PHP class using inheritance. Define a base class `Person` and a derived class `Employee` with additional properties.
- **Q3:** Write a PHP script to demonstrate the use of interfaces.
- **Q4:** Implement a PHP singleton class.
- **Q5:** Write a PHP program that demonstrates method overriding.

4. PHP Database Integration

- **Q1:** Create a PHP script that connects to a MySQL database and fetches all records from a `users` table.
- **Q2:** Write a PHP script using prepared statements to prevent SQL injection while inserting data into a MySQL database.
- **Q3:** Write a PHP script that updates a user's email in the database.
- **Q4:** Write a program to delete a record from a table based on user input.
- **Q5:** Create a registration form with fields for name, email, and password. Store the data in a MySQL database using PHP.

Laravel: Theory Interview Questions

1. Laravel Basics

- **Q1:** What is Laravel, and why is it a popular framework for PHP development?
- **Q2:** Explain the MVC (Model-View-Controller) architecture used in Laravel.
- **Q3:** What is a service provider in Laravel?
- **Q4:** How does routing work in Laravel?
- **Q5:** What are facades in Laravel?

2. Laravel Configuration and Setup

- **Q1:** How do you create a new Laravel project?
- **Q2:** What is the `.env` file in Laravel, and what is its significance?
- **Q3:** What is the purpose of `artisan` in Laravel?

- **Q4:** How does Laravel handle environment-based configuration?
- **Q5:** What are migrations in Laravel, and how are they used?

3. Laravel Database and Eloquent

- **Q1:** Explain what Eloquent ORM is and how it simplifies database interactions in Laravel.
- **Q2:** How can you define relationships (one-to-one, one-to-many, many-to-many) in Eloquent?
- **Q3:** What is a query builder in Laravel?
- **Q4:** What is the difference between `pluck()` and `select()` in Eloquent?
- **Q5:** How do you handle database transactions in Laravel?

4. Authentication and Middleware

- **Q1:** What is middleware in Laravel, and how is it used?
- **Q2:** Explain how Laravel handles user authentication.
- **Q3:** What are guards in Laravel authentication?
- **Q4:** How do you create and register middleware in Laravel?
- **Q5:** What is the `csrf_token()` function in Laravel, and why is it important?

Laravel: Practical Interview Questions

1. Laravel Basics

- **Q1:** Create a simple Laravel project and set up routing to display "Hello, World!" on the homepage.
- **Q2:** Write a Laravel controller that returns a view with dynamic data passed from the controller.
- **Q3:** Create a route in Laravel that accepts a parameter from the URL and displays it on the page.
- **Q4:** Set up a form in Laravel that accepts user input and displays the submitted data.
- **Q5:** Create a middleware that checks if a user is logged in before accessing a certain route.

2. Laravel Database and Eloquent

- **Q1:** Create a migration to set up a `users` table in Laravel with fields for `name`, `email`, and `password`.
- **Q2:** Write a Laravel model for the `User` table and fetch all users from the database using Eloquent.
- **Q3:** Implement a one-to-many relationship between `User` and `Posts` in Laravel.
- **Q4:** Create a query to retrieve all posts by a specific user in Laravel.
- **Q5:** Write a Laravel seed file to populate the `users` table with dummy data.

3. Authentication and Middleware

- **Q1:** Set up user authentication in Laravel using the built-in authentication system (`php artisan make:auth`).

- **Q2:** Create a middleware in Laravel to ensure only authenticated users can access certain pages.
- **Q3:** Implement a custom middleware that checks for an admin role before allowing access to the admin panel.
- **Q4:** Set up a registration form with validation in Laravel and store the user data in the database.
- **Q5:** Create a Laravel project where users can log in and log out using session management.

4. Advanced Laravel Features

- **Q1:** Create a CRUD application in Laravel for managing a list of products.
- **Q2:** Integrate a payment gateway like PayPal into a Laravel project.
- **Q3:** Create a RESTful API using Laravel to handle `GET`, `POST`, `PUT`, and `DELETE` requests.
- **Q4:** Set up Laravel's email functionality to send a confirmation email upon user registration.
- **Q5:** Deploy a simple Laravel application to a live server using GitHub and Laravel Forge.