

PHP FULL STACK

Basic-Level PHP Full Stack Interview Questions (1-25)

1. What is PHP?

- PHP (Hypertext Preprocessor) is a **server-side scripting language** used for web development.

2. What are the main features of PHP?

- Open-source, cross-platform, interpreted, object-oriented, and embedded in HTML.

3. How do you declare a PHP variable?

```
$name = "John";
```

4. What are PHP data types?

- String, Integer, Float, Boolean, Array, Object, NULL, Resource.

5. How do you print output in PHP?

```
echo "Hello World!";  
print "Hello World!";
```

6. What is the difference between echo and print?

- `echo` can output multiple values, `print` returns 1 and is slower.

7. How do you concatenate strings in PHP?

```
$str = "Hello" . " World!";
```

8. What are superglobal variables in PHP?

- `$_GET`, `$_POST`, `$_REQUEST`, `$_SESSION`, `$_COOKIE`, `$_FILES`,
`$_SERVER`, `$_ENV`, `$_GLOBALS`

9. How do you define a constant in PHP?

```
define("SITE_NAME", "MyWebsite");
```

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

- == compares values, === compares values and data types.

11. How do you create an array in PHP?

```
$arr = array("Apple", "Banana", "Cherry");
```

12. What is the difference between indexed and associative arrays?

- Indexed arrays use numeric keys (`$arr[0]`), associative arrays use named keys (`$arr["name"]`).

13. How do you loop through an array in PHP?

```
foreach ($arr as $value) {  
    echo $value;  
}
```

14. What is the difference between include and require?

- `require` stops execution on failure, `include` issues a warning.

15. How do you handle form submission in PHP?

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $name = $_POST["name"];  
}
```

16. How do you start a session in PHP?

```
session_start();  
$_SESSION["user"] = "John";
```

17. How do you set and retrieve cookies in PHP?

```
setcookie("user", "John", time() + 3600);  
echo $_COOKIE["user"];
```

18. How do you connect to a MySQL database using PHP?

```
$conn = mysqli_connect("localhost", "root", "", "test_db");
```

19. How do you fetch data from a database in PHP?

```
$result = mysqli_query($conn, "SELECT * FROM users");  
while ($row = mysqli_fetch_assoc($result)) {  
    echo $row["name"];  
}
```

20. What is the purpose of `mysqli_real_escape_string()`?

- Prevents SQL injection by escaping special characters.

21. What is the difference between GET and POST methods?

- GET is visible in the URL, POST is not.

22. What is CSRF and how do you prevent it?

- Cross-Site Request Forgery, prevented using CSRF tokens.

23. How do you redirect a user in PHP?

```
header("Location: home.php");  
exit();
```

24. What is file handling in PHP?

- Reading, writing, opening, and closing files.

25. How do you upload a file in PHP?

```
move_uploaded_file($_FILES["file"]["tmp_name"], "uploads/" . $_FILES["file"]["name"]);
```

Here are the **Intermediate-Level PHP Full Stack Interview Questions (26-50):**

Intermediate-Level PHP Full Stack Interview Questions (26-50)

26. What is the difference between `require_once` and `include_once`?

- `require_once` ensures the file is included only once and stops execution on failure.
- `include_once` includes the file only once but does not stop execution if it fails.

27. What are prepared statements in PHP?

- They prevent SQL injection and improve performance.

```
$stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");  
$stmt->bind_param("s", $email);  
$stmt->execute();
```

28. What is Object-Oriented Programming (OOP) in PHP?

- PHP supports **classes, objects, inheritance, polymorphism, encapsulation, and abstraction.**

29. How do you define a class and object in PHP?

```
class Car {  
    public $color;  
    function setColor($c) {  
        $this->color = $c;  
    }  
}  
$myCar = new Car();
```

30. What is the difference between public, private, and protected access modifiers?

- **Public:** Accessible everywhere.
- **Private:** Accessible only within the class.
- **Protected:** Accessible within the class and its subclasses.

31. What is a constructor and destructor in PHP?

```
class MyClass {  
    function __construct() { echo "Object created"; }  
    function __destruct() { echo "Object destroyed"; }  
}
```

32. How do you implement inheritance in PHP?

```
class ParentClass {  
    protected $name;  
}  
class ChildClass extends ParentClass {  
    function setName($n) { $this->name = $n; }  
}
```

33. What are interfaces in PHP?

- Interfaces define methods that must be implemented by a class.

```
interface Animal {  
    public function makeSound();  
}
```



34. How do you handle exceptions in PHP?

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

35. What is PDO in PHP?

- PDO (PHP Data Objects) is a database connection abstraction layer.

```
$pdo = new PDO("mysql:host=localhost;dbname=test", "root", "");
```

36. What is the difference between **mysqli** and **PDO**?

- **PDO** supports multiple databases, while **mysqli** is MySQL-specific.

37. What is a singleton design pattern?

```
class Singleton {  
    private static $instance;  
    private function __construct() {}  
    public static function getInstance() {  
        if (!self::$instance) self::$instance = new Singleton();  
        return self::$instance;  
    }  
}
```

38. What are sessions in PHP, and how do you destroy them?

```
session_start();  
session_destroy();
```

39. How do you generate a random string in PHP?

```
echo bin2hex(random_bytes(10));
```

40. What is AJAX and how is it used in PHP?

- AJAX allows sending and receiving data asynchronously.

```
$.ajax({ url: "server.php", success: function(data) { console.log(data); } });
```

41. What is a REST API, and how do you create one in PHP?

- A REST API allows communication between systems using HTTP methods.

```
header("Content-Type: application/json");  
echo json_encode(["message" => "Hello"]);
```

42. How do you make a cURL request in PHP?

```
$ch = curl_init("https://api.example.com");  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
$response = curl_exec($ch);
```

43. How do you sanitize user input in PHP?

```
$email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
```

44. How do you send an email using PHP?

```
mail("user@example.com", "Subject", "Message");
```

45. What is Composer in PHP?

- Composer is a dependency manager for PHP.

```
composer require phpmailer/phpmailer
```

46. How do you secure passwords in PHP?

```
$hashedPassword = password_hash("mypassword", PASSWORD_BCRYPT);
```

47. How do you verify a hashed password in PHP?

```
password_verify("mypassword", $hashedPassword);
```

48. What is CSRF protection, and how do you implement it?

- Use CSRF tokens in forms.

```
<input type="hidden" name="token" value="<?= $_SESSION['token']; ?>">
```

49. What is XSS, and how do you prevent it in PHP?

- Use `htmlspecialchars()`.

```
echo htmlspecialchars($userInput);
```

50. What is the `htaccess` file used for?

- Used for URL rewriting, security, and redirects.

```
RewriteEngine On  
RewriteRule ^about$ about.php [L]
```

Here are the **Advanced-Level PHP Full Stack Interview Questions (51-75):**

Advanced-Level PHP Full Stack Interview Questions (51-75)

51. How do you create a custom PHP framework from scratch?

- Implement **routing, controllers, models, views, and database abstraction**.
- Use **autoloading (Composer), dependency injection, and MVC pattern**.

52. What is the MVC architecture in PHP?

- **Model:** Handles database interactions.
- **View:** Manages UI.
- **Controller:** Manages request logic.

53. How do you implement routing in a PHP framework?

```
$uri = trim($_SERVER['REQUEST_URI'], '/');  
if ($uri == "home") { include "views/home.php"; }
```

54. What is an ORM, and why use it in PHP?

- ORM (Object-Relational Mapping) allows database queries using objects.
- **Example:** Laravel Eloquent ORM.

```
$users = User::where("active", 1)->get();
```

55. How do you handle file uploads securely in PHP?

- Validate file type (`mime_content_type()`).
- Limit file size.
- Store files outside the web root.

56. What is dependency injection in PHP?

- Injecting dependencies into a class instead of hardcoding them.

57. How do you use namespaces in PHP?

```
namespace MyApp\Models;  
use MyApp\Controllers\Controller;
```

58. How do you prevent SQL injection in PHP?

- Use **prepared statements** and **input validation**.

59. What is memcached and how does it improve PHP performance?

- Caches database queries and objects in memory.

60. How do you implement caching in PHP?

```
$cache = apcu_fetch("cached_data");  
if (!$cache) { $cache = expensive_function(); apcu_store("cached_data", $cache); }
```

61. How do you create a REST API in PHP?

- Use `header("Content-Type: application/json");`
- Handle different request methods: GET, POST, PUT, DELETE.

62. What is the difference between REST and SOAP?

- **REST** uses JSON over HTTP, **SOAP** uses XML with strict standards.

63. How do you implement JWT authentication in PHP?

```
$jwt = JWT::encode(["user" => $userId], "secret_key", "HS256");
```

64. What is the difference between horizontal and vertical scaling in PHP applications?

- **Horizontal:** Add more servers.
- **Vertical:** Increase server resources.

65. What are closures in PHP?

```
$greet = function($name) { return "Hello " . $name; };
```

66. How do you handle real-time applications in PHP?

- Use **WebSockets**, **Socket.io**, or **SSE (Server-Sent Events)**.

67. How do you prevent session hijacking in PHP?

- Use **HTTPS**, **regenerate session ID**, and **HttpOnly cookies**.

68. What is the use of **ob_start()** in PHP?

- It enables output buffering.

69. How do you create a custom error handler in PHP?

```
function customErrorHandler($errno, $errstr) { echo "Error: $errstr"; }  
set_error_handler("customErrorHandler");
```

70. How do you integrate payment gateways like Stripe in PHP?

```
\Stripe\Stripe::setApiKey("sk_test_...");  
$charge = \Stripe\Charge::create(["amount" => 5000, "currency" => "usd"]);
```

71. How do you optimize database queries in PHP?

- Use **indexes**, **pagination**, and **limit columns fetched**.

72. How do you create custom middleware in PHP?

```
class AuthMiddleware {  
    public function handle($request, $next) {  
        if (!isset($_SESSION['user'])) die("Unauthorized");  
        return $next($request);  
    }  
}
```

73. What is Laravel, and why is it popular?

- Laravel is a PHP framework with **built-in authentication, ORM, routing, and middleware**.

74. How do you deploy a PHP application to AWS?

- Use **EC2, Elastic Beanstalk, or Lambda**.
- Store files on **S3** and use **RDS** for databases.

75. How do you set up auto-scaling in PHP applications?

- Use **load balancers, Docker, and Kubernetes**.

Here are the **Technical-Level PHP Full Stack Interview Questions (76-100)**:

Technical-Level PHP Full Stack Interview Questions (76-100)

76. How do you set up a LAMP stack for PHP development?

- Install **Linux, Apache, MySQL, PHP**.
- Configure Apache virtual hosts and enable PHP modules.

77. What is the difference between Apache and Nginx?

- **Apache**: Thread-based, supports **.htaccess**.
- **Nginx**: Event-driven, faster for static content.

78. How do you use Redis in PHP?

- Redis is an **in-memory key-value store** for caching.

```
$redis = new Redis();
```

```
$redis->connect("127.0.0.1", 6379);
```

```
$redis->set("user", "John");
```

79. What is the purpose of **.htaccess** in PHP?

- URL rewriting, security settings, and redirections.

```
RewriteEngine On
```

```
RewriteRule ^about$ about.php [L]
```

80. How do you configure a PHP application using environment variables?

- Use **.env** files and **getenv()**.

```
putenv("DB_HOST=localhost");
```

```
echo getenv("DB_HOST");
```

81. How do you implement GraphQL in PHP?

- Use libraries like **Webonyx/GraphQL-PHP**.

```
$queryType = new ObjectType([
```

```
    "name" => "Query",
```

```
    "fields" => ["message" => ["type" => Type::string(), "resolve" => fn() => "Hello!"]]
```

```
]);
```

82. How do you handle long-running PHP scripts?

- Increase **max_execution_time** in **php.ini**.

```
ini_set("max_execution_time", 300);
```

83. How do you handle background jobs in PHP?

- Use **cron jobs** or **queue systems** (Laravel Queue, RabbitMQ, Redis).

```
* * * * * php /var/www/html/script.php
```

84. How do you secure API keys in PHP applications?

- Store them in **environment variables** or a **secure vault**.

85. How do you rate limit API requests in PHP?

- Use **Redis** to track requests per IP.

86. What is the difference between synchronous and asynchronous PHP?

- **Synchronous:** PHP executes line by line.
- **Asynchronous:** Non-blocking execution using libraries like **ReactPHP**.

87. How do you implement OAuth authentication in PHP?

- Use **OAuth 2.0 providers** like Google, Facebook, or Firebase.

88. How do you set up Docker for a PHP application?

- Create a **Dockerfile** and use **docker-compose**.

```
FROM php:8.1-apache
```

```
COPY . /var/www/html/
```

89. How do you use WebSockets in PHP for real-time applications?

- Use **Ratchet PHP WebSocket library**.

90. What is PHP-FPM, and how does it improve performance?

- PHP-FPM (FastCGI Process Manager) handles high-traffic PHP applications.

91. How do you scale PHP applications horizontally?

- Use **load balancers (NGINX, AWS ELB)** and **stateless sessions (Redis, Memcached)**.

92. How do you integrate a microservices architecture with PHP?

- Use **Docker, REST APIs, or gRPC**.

93. What is a headless CMS, and how does PHP integrate with it?

- **Headless CMS** like Strapi or Contentful delivers content via API to PHP applications.

94. How do you set up a CDN for PHP static files?

- Use **Cloudflare, AWS CloudFront, or Akamai**.

95. How do you implement logging in PHP?

- Use **Monolog library** for structured logging.

```
$log = new Logger("app");
```

```
$log->pushHandler(new StreamHandler("app.log", Logger::WARNING));
```

96. How do you monitor PHP application performance?

- Use **New Relic, Blackfire, or Prometheus**.

97. What is horizontal partitioning (sharding) in PHP databases?

- Distributing large databases across multiple servers.

98. How do you deploy a PHP application on Kubernetes?

- Create **Docker images** and deploy using **Helm charts**.

99. How do you prevent directory traversal attacks in PHP?

- Sanitize file paths and restrict file access.

```
$filename = basename($_GET['file']);
```

100. What are some PHP security best practices?

- Use **prepared statements**, **HTTPS**, **CSRF tokens**, and **input validation**.
-

