

PHP interview questions and answers

Basic PHP Questions

1. What is PHP?

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

2. What are the features of PHP?

Answer:

- Open-source
- Platform-independent
- Supports various databases
- Loosely typed
- Supports OOP

3. How do you declare a variable in PHP?

Answer: `$var_name = "value";`

4. What are the different types of variables in PHP?

Answer:

- **String:** `$name = "John";`
- **Integer:** `$age = 25;`
- **Float:** `$price = 10.99;`
- **Boolean:** `$isValid = true;`
- **Array:** `$arr = [1, 2, 3];`
- **Object:** `$obj = new ClassName();`

5. What is the difference between `==` and `===` in PHP?

Answer:

- `==` checks only value.
- `===` checks value and type.

6. What is the use of `isset()` and `empty()`?

Answer:

- `isset($var)`: Returns `true` if the variable is set and is not `NULL`.
- `empty($var)`: Returns `true` if the variable is empty (e.g., `NULL`, `0`, `""`, `false`).

7. How do you create a constant in PHP?

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

8. What is the difference between `require` and `include`?

Answer:

- `require` stops execution if the file is missing.
- `include` shows a warning but continues execution.

9. What is the difference between `require_once` and `require`?

Answer: `require_once` ensures the file is included only once.

10. What are the different types of errors in PHP?

Answer:

- **Parse Error** – Syntax error
- **Fatal Error** – Calling undefined function
- **Warning** – Include missing file
- **Notice** – Using an undefined variable

Intermediate PHP Questions (31-70)

11. What is the use of `explode()` and `implode()`?

Answer:

- `explode(" ", "Hello World")` → `["Hello", "World"]`
- `implode("-", ["Hello", "World"])` → `"Hello-World"`

12. How do you handle exceptions in PHP?

Answer:

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

```
}
```

13. How do you connect MySQL with PHP?

Answer:

```
$conn = new mysqli("localhost", "root", "password", "database");  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}
```

14. What are sessions and cookies?

Answer:

- **Session:** Stores data on the server. (`$_SESSION`)
- **Cookie:** Stores data in the browser. (`setcookie()`)

15. What is the difference between **foreach** and **while** loops?

Answer:

- **foreach** loops through an array.
- **while** runs until a condition is met.

16. How to prevent SQL Injection in PHP?

Answer: Use **prepared statements**:

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

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

Answer:

- **GET:** Sends data in the URL (less secure).
- **POST:** Sends data in the request body (more secure).

18. How do you upload a file in PHP?

Answer:

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

19. What is an associative array?

Answer:

```
$person = ["name" => "John", "age" => 25];  
echo $person["name"];
```

20. How to remove duplicate values from an array?

Answer: `array_unique($array);`

Advanced PHP Questions (71-100)

21. What is the difference between abstract class and interface?

Answer:

- **Abstract class:** Can have method implementations.
- **Interface:** Cannot have method implementations.

22. How to create a class in PHP?

Answer:

```
class Car {  
    public $brand;  
    function __construct($brand) {  
        $this->brand = $brand;  
    }  
}  
$car = new Car("Toyota");
```

23. What is Laravel?

Answer: Laravel is a PHP framework for web applications with MVC architecture.

24. How to check if an array contains a value?

Answer: `in_array("apple", $fruits);`

25. What is JSON in PHP?

Answer: JSON is a format for exchanging data.

- Encode: `json_encode($array);`
- Decode: `json_decode($json, true);`

26. What are magic methods in PHP?

Answer: Special methods starting with `__`, like `__construct()`, `__destruct()`, `__get()`, `__set()`.

27. How do you use traits in PHP?

Answer:

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

28. How do you generate a random number in PHP?

Answer: `rand(1, 100);`

29. How do you reverse a string in PHP?

Answer: `strrev("Hello");`

30. How do you check if a string contains a substring?

Answer: `strpos("Hello World", "World") !== false;`

31: Write a PHP script to find the factorial of a number.

```
function factorial($num) {  
    return ($num == 0) ? 1 : $num * factorial($num - 1);  
}  
echo factorial(5);
```

32: How do you swap two variables without using a third variable?

`$a = 5; $b = 10;`

```
list($a, $b) = [$b, $a];
```

33: Write a function to check if a string is a palindrome.

```
function isPalindrome($str) {  
    return strrev($str) === $str;  
}
```

34: How do you sort an array in PHP?

```
sort($arr);
```

35: How do you get the current date and time in PHP?

```
echo date("Y-m-d H:i:s");
```

36. How do you count elements in an array?

Answer: `count($array);`

37. What is a PHP namespace?

Answer: Used to avoid name conflicts.

```
namespace MyProject;  
class MyClass {}
```

38. How do you send an email in PHP?

Answer: Using `mail()`

```
mail("test@example.com", "Subject", "Message", "From: sender@example.com");
```

39. What are the different types of loops in PHP?

Answer:

- `for`
- `while`
- `do-while`
- `foreach`

40. How do you escape special characters in PHP?

Answer: Use `addslashes()` or `htmlspecialchars()`.

```
$str = addslashes("John's book"); // Output: John\'s book
```

41. What is a static variable in PHP?

Answer: Retains its value between function calls.

```
function counter() {  
    static $count = 0;  
    return ++$count;  
}
```

42. What is the purpose of `serialize()` and `unserialize()`?

Answer: Converts objects/arrays to a storable format.

```
$data = serialize(["name" => "John"]);  
$original = unserialize($data);
```

43. How do you get the IP address of a user?

Answer: `$_SERVER['REMOTE_ADDR']`

44. What is the difference between `header()` and `setcookie()`?

Answer:

- `header()` sends HTTP headers.
- `setcookie()` stores data in cookies.

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

Answer:

```
echo bin2hex(random_bytes(10)); // Generates a 20-character string
```

46. What is the purpose of `json_encode()` and `json_decode()`?

Answer: Convert JSON data.

```
$json = json_encode(["name" => "John"]);  
$data = json_decode($json, true);
```

47. How do you create a session in PHP?

Answer:

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

48. How do you destroy a session?

Answer:

```
session_destroy();
unset($_SESSION['user']);
```

49. What is method chaining in PHP?

Answer: Calling multiple methods on an object in a single statement.

```
class Test {
    function hello() { echo "Hello "; return $this; }
    function world() { echo "World!"; return $this; }
}
$obj = new Test();
$obj->hello()->world(); // Output: Hello World!
```

50. How do you perform file handling in PHP?

Answer: Using `fopen()`, `fwrite()`, `fread()`, and `fclose()`.

```
$file = fopen("file.txt", "w");
fwrite($file, "Hello, PHP!");
fclose($file);
```

Advanced PHP Questions (71-100)

71. How do you implement inheritance in PHP?

Answer:

```
class ParentClass {
    public function greet() { echo "Hello"; }
}
class ChildClass extends ParentClass {}
```

72. What is polymorphism in PHP?

Answer: Objects of different classes sharing the same interface.

73. What is a constructor in PHP?

Answer: `__construct()` is called automatically when an object is created.

```
class Car {  
    function __construct() { echo "Car created"; }  
}  
$car = new Car();
```

74. How do you use a singleton pattern in PHP?

Answer:

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

TECHPARK
GUIDE'S FOR PERFECT CAREER PATHWAY

75. What is dependency injection?

Answer: Injecting dependencies into a class instead of creating them inside the class.

76. What is an interface in PHP?

Answer:

```
interface Animal {  
    public function makeSound();  
}  
class Dog implements Animal {  
    public function makeSound() { echo "Bark"; }  
}
```

77. How do you handle large file uploads in PHP?

Answer: Increase `upload_max_filesize` and `post_max_size` in `php.ini`.

78. How to secure PHP applications?

Answer:

- Use prepared statements
- Escape user input
- Validate user input
- Use HTTPS

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

Answer: Using `json_encode()`, `$_GET`, and `$_POST`.

80. What is Composer in PHP?

Answer: Dependency manager for PHP projects.

Coding Exercises (81-100)

81. Reverse an array without using `array_reverse()`.

```
function reverseArray($arr) {  
    $rev = [];  
    for ($i = count($arr) - 1; $i >= 0; $i--) {  
        $rev[] = $arr[$i];  
    }  
    return $rev;  
}
```

82. Find the Fibonacci series up to `n`.

```
function fibonacci($n) {  
    $a = 0; $b = 1;  
    for ($i = 0; $i < $n; $i++) {  
        echo $a . " ";  
        $temp = $a + $b;  
        $a = $b;  
        $b = $temp;  
    }  
}
```

83. Check if a number is prime.

```
function isPrime($num) {
    if ($num < 2) return false;
    for ($i = 2; $i <= sqrt($num); $i++) {
        if ($num % $i == 0) return false;
    }
    return true;
}
```

Here are **PHP interview questions 84-100**, covering advanced concepts and coding exercises.

Advanced PHP Questions (84-100)

84. What is a closure in PHP?

Answer: A closure is an anonymous function that can capture variables from its surrounding scope.

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

```
$greet = function() use ($name) {
    echo "Hello, $name";
};
```

```
$greet(); // Output: Hello, John
```

85. How do you use **array_reduce()** in PHP?

Answer: **array_reduce()** applies a function to elements of an array and returns a single value.

```
$numbers = [1, 2, 3, 4, 5];
```

```
$sum = array_reduce($numbers, fn($carry, $item) => $carry + $item, 0);
```

```
echo $sum; // Output: 15
```

86. What is SPL (Standard PHP Library)?

Answer: SPL is a collection of interfaces and classes that provide standard data structures, iterators, and exceptions.

87. How do you create a generator in PHP?

Answer: A generator allows you to iterate over large datasets efficiently.

```
function generateNumbers() {  
    for ($i = 1; $i <= 5; $i++) {  
        yield $i;  
    }  
}  
  
foreach (generateNumbers() as $num) {  
    echo $num . " ";  
}  
// Output: 1 2 3 4 5
```

88. What is late static binding in PHP?

Answer: Late static binding refers to the use of `static::` instead of `self::` to refer to the called class.

```
class ParentClass {  
    public static function who() {  
        echo static::class;  
    }  
}  
  
class ChildClass extends ParentClass {}  
  
ChildClass::who(); // Output: ChildClass
```

89. What is the difference between `$this` and `self::`?

Answer:

- `$this` refers to the current object instance.
- `self::` refers to the class itself (used for static methods).

90. What is PSR in PHP?

Answer: PHP Standards Recommendations (PSR) define coding standards for PHP frameworks and libraries (e.g., PSR-1, PSR-2, PSR-4).

91. What is the difference between `__autoload()` and `spl_autoload_register()`?

Answer:

- `__autoload()` is deprecated.
- `spl_autoload_register()` allows multiple autoload functions.

```
spl_autoload_register(function($class) {  
    include $class . '.php';  
});
```

92. How do you implement an iterator in PHP?

Answer: Using the `Iterator` interface.

```
class NumberIterator implements Iterator {  
    private $numbers = [1, 2, 3];  
    private $index = 0;  
  
    public function current() { return $this->numbers[$this->index]; }  
    public function key() { return $this->index; }  
    public function next() { $this->index++; }  
    public function rewind() { $this->index = 0; }  
    public function valid() { return isset($this->numbers[$this->index]); }  
}  
  
$iterator = new NumberIterator();  
foreach ($iterator as $value) {  
    echo $value . " ";  
}
```

```
}
```

93. How do you handle errors in PHP using error handling functions?

Answer:

```
set_error_handler(function($errno, $errstr) {  
    echo "Error: [$errno] $errstr";  
});  
  
trigger_error("Custom error", E_USER_WARNING);
```

94. What is the difference between `__get()` and `__set()` in PHP?

Answer: Magic methods to handle inaccessible properties.

```
class Test {  
    private $data = [];  
    public function __get($key) { return $this->data[$key] ?? null; }  
    public function __set($key, $value) { $this->data[$key] = $value; }  
}  
  
$obj = new Test();  
  
$obj->name = "John";  
  
echo $obj->name; // Output: John
```

95. What are coroutines in PHP?

Answer: Coroutines allow non-blocking code execution using `yield`.

```
function task() {  
    echo "Step 1\n";  
    yield;  
    echo "Step 2\n";  
}
```

```

}

$gen = task();

$gen->current();

$gen->next();

```

96. What is the difference between `session_unset()` and `session_destroy()`?

Answer:

- `session_unset()` clears session variables but keeps the session active.
- `session_destroy()` completely removes the session.

97. How do you create an event listener in PHP?

Answer:

```

class Event {
    private $listeners = [];

    public function on($event, $callback) { $this->listeners[$event][] = $callback; }

    public function trigger($event, ...$args) {
        foreach ($this->listeners[$event] ?? [] as $callback) {
            call_user_func($callback, ...$args);
        }
    }
}

$event = new Event();

$event->on("login", fn($user) => echo "$user logged in");

$event->trigger("login", "John");

```

98. How do you optimize a PHP application for performance?

Answer:

- Use **OPcache**
- Minimize database queries
- Use **CDN**
- Enable **Gzip compression**
- Optimize loops and algorithms

99. How do you use WebSockets in PHP?

Answer: Using [Ratchet](#) or [swoole](#).

```
use Ratchet\MessageComponentInterface;
```

```
use Ratchet\ConnectionInterface;
```

```
class Chat implements MessageComponentInterface {  
  
    public function onOpen(ConnectionInterface $conn) { echo "New connection\n"; }  
  
    public function onMessage(ConnectionInterface $conn, $msg) { echo "Message: $msg\n";  
}  
  
    public function onClose(ConnectionInterface $conn) { echo "Connection closed\n"; }  
  
    public function onError(ConnectionInterface $conn, \Exception $e) { echo "Error: " .  
$e->getMessage(); }  
  
}
```

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

Answer:

- **Synchronous:** Code executes sequentially.
- **Asynchronous:** Tasks run in parallel using tools like [swoole](#) or [ReactPHP](#).