

45 Advanced PHP Interview Questions That May Land You a Job



Join Our [Referral
\\$](#)

[Program](#), Share And Get Paid

The average PHP Developer salary in Australia is \$95,000 per year or \$48.72 per hour. Entry level positions start at \$55,000 per year while most experienced workers make up to \$161,500 per year. Follow along and learn 45 advanced PHP interview questions that may land you a job.

Q1:

How can you pass a variable by reference?

Entry



[PHP 82](#)

Answer

To be able to pass a variable by **reference**, we use an *ampersand* in front of it, as follows:

`$var1 = &$var2`

Having Tech or Coding Interview? Check [82 PHP Interview Questions](#)

Source: [guru99.com](#)

Q2:

What does **\$GLOBALS** mean?

Entry



PHP 82

Answer

\$GLOBALS is associative array including references to all variables which are currently defined in the global scope of the script.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [guru99.com](#)

Q3:

What is the difference between == and
===?

Entry

**PHP 82**

Answer

- The operator == casts between two different types if they are different
- The === operator performs a '*typesafe comparison*'

That means that it will only return true if both operands have the same type and the same value.

```
1 === 1: true
1 == 1: true
1 === "1": false // 1 is an integer, "1" is a string
1 == "1": true // "1" gets casted to an integer, which is 1
"foo" === "foo": true // both operands are strings and have the same value
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q4:

What is the use of ini_set()?

Entry

**PHP 82**

Answer

PHP allows the user to modify some of its settings mentioned in php.ini using ini_set(). This function requires two string arguments. First one is the name of the setting to be modified and the second one is the new value to be assigned to it.

Given line of code will enable the display_error setting for the script if it's disabled.

```
ini_set('display_errors', '1');
```

We need to put the above statement, at the top of the script so that, the setting remains enabled till the end. Also, the values set via ini_set() are applicable, only to the current script. Thereafter, PHP will start using the original values from php.ini.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: github.com/Bootsy

Q6:

Declare some function with default parameter

Junior



[PHP 82](#)

Answer

Consider:

```
function showMessage($hello = false){
    echo ($hello) ? 'hello' : 'bye';
}
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: codementor.io

Q7:

Differentiate between echo and print()

Junior



[PHP 82](#)

Answer

`echo` and `print` are more or less the same. They are both used to output data to the screen.

The differences are:

- `echo` has no return value while `print` has a return value of 1 so it can be used in expressions.
- `echo` can take multiple parameters (although such usage is rare) while `print` can take one argument.
- `echo` is faster than `print`.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: github.com/Bootsy

Q8:**Explain how we handle exceptions in PHP?**

Junior

**[PHP 82](#)****Answer**

When an exception is thrown, code following the statement will not be executed, and PHP will attempt to find the first matching catch block. If an exception is not caught, a PHP Fatal Error will be issued with an "Uncaught Exception". An exception can be thrown, and caught within PHP.

To handle exceptions, code may be surrounded in a `try` block. Each `try` must have at least one corresponding `catch` block. Multiple `catch` blocks can be used to catch different classes of exceptions. Exceptions can be thrown (or re-thrown) within a `catch` block.

Consider:

```
try {
    print "this is our try block n";
    throw new Exception();
} catch (Exception $e) {
    print "something went wrong, caught yah! n";
} finally {
    print "this part is always executed n";
}
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: github.com/Bootsy



React 155



ASP.NET Web API 33



Software Testing 26



Ruby on Rails 72



TypeScript 39



Reactive Programming 12



DevOps 44



GraphQL 25



Agile & Scrum 47



HTML5 55

Q9:

Explain what the different PHP errors are

Junior



PHP 82

Answer

- A **notice** is a non-critical error saying something went wrong in execution, something minor like an undefined variable.
- A **warning** is given when a more critical error like if an `include()` command went to retrieve a non-existent file. In both this and the error above, the script would continue.
- A **fatal error** would terminate the code. Failure to satisfy a `require()` would generate this type of error, for example.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [pangara.com](#)

[Top 29+ Advanced Blockchain Interview Questions \(ANSWERED\)](#)

(B)

[Blockchain 42](#)

Q10:

How can you enable error reporting in PHP?

Junior



[PHP 82](#)

Answer

Check if “`display_errors`” is equal “on” in the `php.ini` or declare “`ini_set('display_errors', 1)`” in your script.

Then, include “`error_reporting(E_ALL)`” in your code to display all types of error messages during the script execution.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [codementor.io](#)

Q11:

In PHP, objects are they passed by value or by reference?

Junior



[PHP 82](#)

Answer

In PHP, objects passed by **value**.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [guru99.com](#)

Q12:

What are the differences between die() and exit() functions in PHP?

Junior



[PHP 82](#)

Answer

There's no difference - they are the same. The only advantage of choosing die() over exit(), might be the time you spare on typing an extra letter.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q13:

What are the main differences between const vs define

Junior



[PHP 82](#)

Answer

The fundamental difference between const vs define is that const defines constants at compile time, whereas define defines them at run time.

```
const FOO = 'BAR';
define('FOO', 'BAR');

// but
if (...) {
    const FOO = 'BAR';      // Invalid
}
```

```
if (...) {
    define('FOO', 'BAR'); // Valid
}
```

Also until PHP 5.3, `const` could not be used in the global scope. You could only use this from within a class. This should be used when you want to set some kind of constant option or setting that pertains to that class. Or maybe you want to create some kind of enum. An example of good `const` usage is to get rid of magic numbers.

`Define` can be used for the same purpose, but it can only be used in the global scope. It should only be used for global settings that affect the entire application.

Unless you need any type of conditional or expressional definition, use `consts` instead of `define()`- simply for the sake of readability!

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

[33+ Node.js Interview Questions \(ANSWERED\) Devs Should Know in 2020](#)



[Node.js 88](#)

Q14:

What is PDO in PHP?

Junior



[PHP 82](#)

Answer

PDO stands for PHP Data Object.

It is a set of PHP extensions that provide a core PDO class and database, specific drivers. It provides a vendor-neutral, lightweight, data-access abstraction layer. Thus, no matter what database we use, the function to issue queries and fetch data will be same. It focuses on data access abstraction rather than database abstraction.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [github.com/Bootsy](#)

Q15:

What is stdClass in PHP?

Junior

**PHP 82**

Answer

`stdClass` is just a generic 'empty' class that's used when casting other types to objects. `stdClass` is **not** the base class for objects in PHP. This can be demonstrated fairly easily:

```
class Foo{}  
$foo = new Foo();  
echo ($foo instanceof stdClass)?'Y':'N'; // outputs 'N'  
It is useful for anonymous objects, dynamic properties, etc.
```

An easy way to consider the `StdClass` is as an alternative to associative array. See this example below that shows how `json_decode()` allows to get an `StdClass` instance or an associative array. Also but not shown in this example, `SoapClient::__soapCall` returns an `StdClass` instance.

```
//Example with StdClass  
$json = '{ "foo": "bar", "number": 42 }';  
$stdInstance = json_decode($json);  
  
echo $stdInstance -> foo.PHP_EOL; //"bar"  
echo $stdInstance -> number.PHP_EOL; //42
```

```
//Example with associative array  
$array = json_decode($json, true);
```

```
echo $array['foo'].PHP_EOL; //"bar"  
echo $array['number'].PHP_EOL; //42
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

**ASP.NET Web API 33****C# 115**

[PWA 22](#)



[TypeScript 39](#)



[Ruby 84](#)



[Ruby on Rails 72](#)



[OOP 54](#)



[Software Testing 26](#)



[Golang 49](#)



[Agile & Scrum 47](#)

Q16:

What is the difference between `var_dump()` and `print_r()`?

Junior



[PHP 82](#)

Answer

- The `var_dump` function displays structured information about variables/expressions including its **type** and **value**. Arrays are explored recursively with values indented to show structure. It also shows which array values and object properties are references.
- The `print_r()` displays information about a variable in a way that's readable by humans. array values will be presented in a format that shows keys and elements. Similar notation is used for objects.

Consider:

```
$obj = (object) array('qualitypoint', 'technologies', 'India');
var_dump($obj) will display below output in the screen:
```

```
object(stdClass)#1 (3) {
    [0]=> string(12) "qualitypoint"
    [1]=> string(12) "technologies"
    [2]=> string(5) "India"
}
```

And, print_r(\$obj) will display below output in the screen.

```
stdClass Object (
    [0] => qualitypoint
    [1] => technologies
    [2] => India
)
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q18:

What's the difference between isset() and
array_key_exists()?

Junior



[PHP 82](#)

Answer

- array_key_exists will tell you if a key exists in an array and complains when \$a does not exist.
- isset will only return true if the key/variable exists **and is not null**.
isset doesn't complain when \$a does not exist.

Consider:

```
$a = array('key1' => 'Foo Bar', 'key2' => null);
```

```
isset($a['key1']);           // true
array_key_exists('key1', $a); // true
```

```
isset($a['key2']);           // false
array_key_exists('key2', $a); // true
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q19:

When should I use require vs. include?

Junior



[PHP 82](#)

Answer

The `require()` function is identical to `include()`, except that it handles errors differently. If an error occurs, the `include()` function generates a warning, but the script will continue execution. The `require()` generates a fatal error, and the script will stop.

My suggestion is to just use `require_once` 99.9% of the time.

Using `require` or `include` instead implies that your code is not **reusable** elsewhere, i.e. that the scripts you're pulling in actually execute code instead of making available a class or some function libraries.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q20:

Check if PHP array is associative

Mid



[PHP 82](#)

Answer

Consider:

```
function has_string_keys(array $array) {
    return count(array_filter(array_keys($array), 'is_string')) > 0;
}
```

If there is at least one string key, `$array` will be regarded as an *associative array*.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q21:

Differentiate between exception and error

Mid



PHP 82

Answer

- Recovering from Error is not possible. The only solution to errors is to terminate the execution. Whereas you can recover from Exception by using either try-catch blocks or throwing exception back to caller.
- You will not be able to handle the Errors using try-catch blocks. Even if you handle them using try-catch blocks, your application will not recover if they happen. On the other hand, Exceptions can be handled using try-catch blocks and can make program flow normal if they happen.
- Exceptions are related to application whereas Errors are related to environment in which application is running.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [github.com/Bootsy](#)

[32 Advanced Data Structures Interview Questions \(ANSWERED\) To Smash Your Next Programming Interview](#)



Data Structures 44

Q22:

Differentiate between parameterised and non parameterised functions

Mid



PHP 82

Answer

- **Non parameterised functions** don't take any parameter at the time of calling.
- **Parameterised functions** take one or more arguments while calling. These are used at run time of the program when output depends on dynamic values given at run time There are two ways to access the parameterised function:

1. *call by value*: (here we pass the value directly)
2. *call by reference*: (here we pass the address location where the value is stored)

Having Tech or Coding Interview? Check ↗ 82 PHP Interview Questions

Source: github.com/Bootsity



[ADO.NET 33](#)



[CSS 50](#)



[.NET Core 52](#)



[Reactive Programming 12](#)



[React Native 72](#)



[LINQ 38](#)



[jQuery 51](#)



[Xamarin 83](#)



[Golang 49](#)



[Angular 120](#)**Q23:****Explain function call by reference**

Mid

[PHP 82](#)

Answer

In case of call by reference, actual value is modified if it is modified inside the function. In such case, we need to use & symbol with formal arguments. The & represents reference of the variable.

Example:

```
function adder(&$str2) {
    $str2 .= 'Call By Reference';
}
$str = 'This is ';
adder($str);
echo $str;
```

Output:

This is Call By Reference

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: github.com/Bootsy

Q24:**Explain the difference between exec() vs system() vs passthru()?**

Mid

[PHP 82](#)

Answer

- exec() is for calling a system command, and perhaps dealing with the output yourself.

- `system()` is for executing a system command and immediately displaying the output - presumably text.
- `passthru()` is for executing a system command which you wish the raw return from - presumably something binary.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)
Source: [stackoverflow.com](#)

Q25:

How do I pass variables and data from PHP to JavaScript?

Mid



PHP 82

Answer

There are actually several approaches to do this:

- Use AJAX to get the data you need from the server.
 Consider **get-data.php**:

```
echo json_encode(42);
```

Consider **index.html**:

```
<script>
    function reqListener () {
        console.log(this.responseText);
    }

    var oReq = new XMLHttpRequest(); // New request object
    oReq.onload = function() {
        // This is where you handle what to do with the response.
        // The actual data is found on this.responseText
        alert(this.responseText); // Will alert: 42
    };
    oReq.open("get", "get-data.php", true);
    //                                         ^ Don't block the rest of the execution.
    //                                         Don't wait until the request finishes to
    //                                         continue.
    oReq.send();
</script>
```

- Echo the data into the page somewhere, and use JavaScript to get the information from the DOM.

```
<div id="dom-target" style="display: none;">
<?php
    $output = "42"; // Again, do some operation, get the output.
    echo htmlspecialchars($output); /* You have to escape because the result
                                    will not be valid HTML otherwise. */
?
</div>
<script>
    var div = document.getElementById("dom-target");
    var myData = div.textContent;
</script>
```

- Echo the data directly to JavaScript.

```
<script>
    var data = <?php echo json_encode("42", JSON_HEX_TAG); ?>; // Don't forget the extra semic
</script>
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

[75+ JavaScript Interview Questions \[ES6/ES2015 Update\] in 2019](#)



[JavaScript 142](#)

Q26:

How would you create a Singleton class
using PHP?

Mid



[PHP 82](#)

Answer

```
/**
 * Singleton class
 *
 */
final class UserFactory {
```

```

    /**
     * Call this method to get singleton
     *
     * @return UserFactory
     */
    public static
    function Instance() {
        static $inst = null;
        if ($inst === null) {
            $inst = new UserFactory();
        }
        return $inst;
    }

    /**
     * Private ctor so nobody else can instantiate it
     *
     */
    private
    function __construct() {

    }
}

```

To use:

```
$fact = UserFactory::Instance();
$fact2 = UserFactory::Instance();
```

But:

```
$fact = new UserFactory()
```

Throws an error.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q27:

Is there a function to make a copy of a PHP array to another?

Mid



PHP 82

Answer

In PHP arrays are assigned by copy, while objects are assigned by reference so PHP will copy the array by default. References in PHP have to be explicit:

```
$a = array(1,2);
$b = $a; // $b will be a different array
$c = &$a; // $c will be a reference to $a
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q28:**What are the exception class functions?**

Mid

**PHP 82**

Answer

There are following functions which can be used from Exception class.

- getMessage() – message of exception
- getCode() – code of exception
- getFile() – source filename
- getLine() – source line
- getTrace() – n array of the backtrace()
- getTraceAsString() – formated string of trace
- Exception::__toString gives the string representation of the exception.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [github.com/Bootsy](#)

Q30:**What is the difference between PDO's query() vs execute()?**

Mid



PHP 82

Answer

- `query` runs a standard SQL statement and requires you to properly escape all data to avoid SQL Injections and other issues.
- `execute` runs a prepared statement which allows you to bind parameters to avoid the need to escape or quote the parameters. `execute` will also perform better if you are repeating a query multiple times.

Best practice is to stick with prepared statements and `execute` for increased security. Aside from the escaping on the client-side that it provides, a *prepared statement* is *compiled* on the server-side once, and then can be passed different parameters at each execution.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q31:

What is use of Null Coalesce Operator?

Mid

**PHP 82**

Answer

Null coalescing operator returns its first operand if it exists and is not NULL. Otherwise it returns its second operand.

Example:

```
$name = $firstName ?? $username ?? $placeholder ?? "Guest";
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [github.com/Bootsy](#)

Q32:

What will be returned by this code?

Mid

**PHP 82**

Problem

Consider the code:

```
$a = new stdClass();
$a->foo = "bar";
$b = clone $a;
var_dump($a === $b);
```

What will be echoed to the console?

Answer

Two instances of the same class with equivalent members do NOT match the `==` operator. So the answer is:

`bool(false)`

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Q33:

What will be returned by this code? Explain the result.

Mid

**PHP 82****Problem**

Consider the code. What will be returned as a result?

```
$something = 0;
echo ('password123' == $something) ? 'true' : 'false';
```

Answer

The answer is `true`. You should never use `==` for string comparison. Even if you are comparing strings to strings, PHP will implicitly cast them to floats and do a numerical comparison if they appear numerical. `==` is OK.

For example

```
'1e3' == '1000' // true
```

also returns true.

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: [stackoverflow.com](#)

Top 29 AngularJS Interview Questions (ANSWERED) You Will Be Asked Tomorrow

Q34:

When should I use require_once vs. require?

Mid



PHP 82

Answer

The `require_once()` statement is identical to `require()` except PHP will check if the file has already been included, and if so, not include (require) it again.

My suggestion is to just use `require_once` 99.9% of the time.

Using `require` or `include` instead implies that your code is not **reusable** elsewhere, i.e. that the scripts you're pulling in actually execute code instead of making available a class or some function libraries.

Having Tech or Coding Interview? Check ↗ 82 PHP Interview Questions

Source: stackoverflow.com

Q35:

Why do we use extract()?

Mid



PHP 82

Answer

The `extract()` function imports variables into the local symbol table from an array. This function uses array keys as variable names and values as variable values. For each element it will create a variable in the current symbol table. This function returns the number of variables extracted on success.

Example:

```
$a = "Original";
$my_array = array("a" => "Cat", "b" => "Dog", "c" => "Horse");
extract($my_array);
echo "\$a = $a; \$b = $b; \$c = $c";
```

Output:

```
$a = Cat; $b = Dog; $c = Horse
```

Having Tech or Coding Interview? Check ↗ [82 PHP Interview Questions](#)

Source: github.com/Bootsy

Q38:

How to measure execution times of PHP scripts?

Senior



[PHP 82](#)

Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

Join 62k+ Developer Who Trust FullStack.Cafe

Q39:

Is PHP single or multi threaded?

Senior



[PHP 82](#)

Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

Join 62k+ Developer Who Trust FullStack.Cafe

Q40:

What exactly are late static bindings in PHP?

Senior



[PHP 82](#)

Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

Join 62k+ Developer Who Trust FullStack.Cafe

Q42:

What is use of Spaceship Operator?

Senior



PHP 82

Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

Join 62k+ Developer Who Trust FullStack.Cafe

Q44:

How could we implement method overloading in PHP?

Expert



PHP 82

Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Great

based on [56 Trustpilot Reviews](#)