PHP Interview Tip #1

“One of the most important things I watch out is the creativity of the person I interview. I try to catch the developers who are constantly learning new things, they are driven by curiosity and also very creative, not just in problem solving but in general too. I mostly ask about PHP and PostgreSQL (I use this combo most of the time).” ~ Laszlo Levente Mári, ex-Googler

PHP Interview Question #1

What’s the difference between the include() and require() functions?

(Question provided by Laszlo Levente Mári)

they both include a specific file but on require the process exits with a fatal error if the file can’t be included, while include statement may still pass and jump to the next step in the execution.

PHP Interview Question #2

How can we get the IP address of the client?

(Question provided by Laszlo Levente Mári)

This question might show an interview how playful and creative the candidate is because there are many options. $\_SERVER["REMOTE\_ADDR"]; is the easiest solution, but you can write x line scripts for this question.

PHP Interview Question #3

What’s the difference between unset() and unlink()

(Question provided by Laszlo Levente Mári)

unset() sets a variable to “undefined” while unlink() deletes a file we pass to it from the file system.

PHP Interview Question #4

What is the output of the following code:

$a = '1';

$b = &$a;

$b = "2$b";

echo $a.", ".$b;

(Question provided by Laszlo Levente Mári)

The answer is 21, 21

Laszlo Levente Mári is an ex-Googler (front-end) and has taken several interview opportunities for fun. SitePoint Author, and full-stack developer at IDG where he is currently building the CMS of PCWorld.com.

Schedule a mock interview with Laszlo!

Want to ace your technical interview? Schedule a Technical Interview Practice Session with an expert now!

PHP Interview Tip #2

“During the interview of a potential candidate I am aiming to understand how updated they are with the new language features as well as their level of understanding of basic operations. In my opinion, this will define how good a developer will become in the future.” ~Agli Pançi, Lead Developer

PHP Interview Question #5

What are the main error types in PHP and how do they differ?

(Question provided by Agli Pançi)

In PHP there are three main type of errors:

Notices – Simple, non-critical errors that are occurred during the script execution. An example of a Notice would be accessing an undefined variable.

Warnings – more important errors than Notices, however the scripts continue the execution. An example would be include() a file that does not exist.

Fatal – this type of error causes a termination of the script execution when it occurs. An example of a Fatal error would be accessing a property of a non-existent object or require() a non-existent file.

Understanding the error types is very important if you are new to programming because they help you understand what is going on during the development, and they will help you know what you should look for in the code during debugging.

PHP Interview Question #6

What is the difference between GET and POST?

(Question provided by Agli Pançi)

– GET displays the submitted data as part of the URL, during POST this information is not shown as it’s encoded in the request.

– GET can handle a maximum of 2048 characters, POST has no such restrictions.

– GET allows only ASCII data, POST has no restrictions, binary data are also allowed.

– Normally GET is used to retrieve data while POST to insert and update.

Understanding the fundamentals of the HTTP protocol is very important to have a good start as a PHP developer, and the differences between GET and POST are an essential part of it.

PHP Interview Question #7

How can you enable error reporting in PHP?

(Question provided by Agli Pançi)

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.

Enabling error messages is very important especially during the debugging process as you can instantly get the exact line that is producing the error and you can see also if the script in general is behaving correctly.

PHP Interview Question #8

What are Traits?

(Question provided by Agli Pançi)

Traits are a mechanism that allows you to create reusable code in languages like PHP where multiple inheritance is not supported. A Trait cannot be instantiated on its own.

It’s important that a developer know the powerful features of the language (s)he is working on, and Trait is one of such features.

PHP Interview Question #9

Can the value of a constant change during the script’s execution?

(Question provided by Agli Pançi)

No, the value of a constant cannot be changed once it’s declared during the PHP execution.

PHP Interview Question #10

Can you extend a Final defined class?

(Question provided by Agli Pançi)

No, you cannot extend a Final defined class. A Final class or method declaration prevents child class or method overriding.

PHP Interview Question #11

What are the \_\_construct() and \_\_destruct() methods in a PHP class?

(Question provided by Agli Pançi)

All objects in PHP have Constructor and Destructor methods built-in. The Constructor method is called immediately after a new instance of the class is being created, and it’s used to initialize class properties. The Destructor method takes no parameters.

Understanding these two in PHP means that the candidate knows the very basics of OOP in PHP.

PHP Interview Question #12

How we can get the number of elements in an array?

(Question provided by Agli Pançi)

The count() function is used to return the number of elements in an array.

Understanding of arrays and array related helper functions it’s a knowledge that every junior developer should have.

Agli Pançi is the Lead Developer of Manoolia and the co-founder of Zero1 Community Albania. Schedule a mock interview with Agli!

PHP Interview Tip #3

“I would expect from a junior developer is that know the basics of programing and how apply these basics to php. I wouldn’t expect that a Junior Developer knows the name of certain functions or complex terms on PHP.” ~Oscar Andrés Monroy, Senior PHP Developer

PHP Interview Question #13

How would you declare a function that receives one parameter name hello?

If hello is true, then the function must print hello, but if the function doesn’t receive hello or hello is false the function must print bye.

(Question provided by Oscar Andrés Monroy Medina)

<?php

function showMessage($hello=false){

echo ($hello)?'hello':'bye';

}

?>

In this question, the interviewer can evaluate if the developer knows how to declare a function and how they would manage the fact of the parameter can or cannot be on the function call. The interviewer can also evaluate if the developer knows the if syntax and if they knows how to print text(echo function).

PHP Interview Question #14

The value of the variable input is a string 1,2,3,4,5,6,7. How would you get the sum of the integers contained inside input?

(Question provided by Oscar Andrés Monroy Medina)

<?php

echo array\_sum(explode(',',$input));

?>

The explode function is one of the most used functions in PHP, so it’s important to know if the developer knows this function. There is no unique answer to this question, but the answer must be similar to this one.

PHP Interview Question #15

Suppose you receive a form submitted by a post to subscribe to a newsletter. This form has only one field, an input text field named email. How would you validate whether the field is empty? Print a message "The email cannot be empty" in this case.

(Question provided by Oscar Andrés Monroy Medina)

<?php

if(empty($\_POST['email'])){

echo "The email cannot be empty";

}

?>

In this question, you will be evaluated on your knowledge about forms management and validation. There is not unique answer for this question, but it must be similar to this one.

PHP Interview Question #15

Suppose that you have to implement a class named Dragonball. This class must have an attribute named ballCount (which starts from 0) and a method iFoundaBall. When iFoundaBall is called, ballCount is increased by one. If the value of ballCount is equal to seven, then the message You can ask your wish is printed, and ballCount is reset to 0. How would you implement this class?

(Question provided by Oscar Andrés Monroy Medina)

<?php

class dragonBall{

private $ballCount;

public function \_\_construct(){

$this->ballCount=0;

}

public function iFoundaBall(){

$this->ballCount++;

if($this->ballCount===7){

echo "You can ask for your wish.";

$this->ballCount=0;

}

}

}

?>

This question will evaluate a candidate’s knowledge about OOP.

Oscar Andrés Monroy has 8 years of experience with PHP. He specializes in PHP, SQL, and data mining. Schedule a mock interview with Oscar!

PHP Interview Question #16

What are the 3 scope levels available in PHP and how would you define them?

(Question provided by Merlyn Coslett)

Private – Visible only in its own class

Public – Visible to any other code accessing the class

Protected – Visible only to classes parent(s) and classes that extend the current class

This is important for any developer to know because it shows an understanding that building applications is more than just being able to write code. One must also have an understanding about privileges and accessibility of that code. There are times protected variables or methods are extremely important, and an understanding of scope is needed to protect the integrity of the data in your application along with provide a clear path through the code.

PHP Interview Question #17

What are getters and setters and why are they important?

(Question provided by Merlyn Coslett)

Getters and setters are methods used to declare or obtain the values of variables, usually private ones. They are important because it allows for a central location that is able to handle data prior to declaring it or returning it to the developer. Within a getter or setter you are able to consistently handle data that will eventually be passed into a variable or additional functions. An example of this would be a user’s name. If you are not using a setter and just declaring the $userName variable by hand you could end up with results as such: "kevin", "KEVIN", "KeViN", "", etc. With a setter you can not only adjust the value, for example, ucfirst($userName), but you can also handle situations where the data is not valid such as the example where "" is passed. The same applies to a getter – when the data is being returned, you can modify the results to include strtoupper($userName) for proper formatting further up the chain.

This is important for any developer who is looking to enter a team-based / application development job to know. Getters and setters are often used when dealing with objects, especially ones that will end up in a database or other storage medium. Because PHP is commonly used to build web applications you will run across getters and setters in more advanced environments, even as a junior developer. They are extremely powerful yet not talked about very much. You can really impress an interviewer by knowing what they are and how to use them early on.

PHP Interview Question #18

What does MVC stand for and what does each component do?

(Question provided by Merlyn Coslett)

MVC stands for Model View Controller.

The controller handles data passed to it by the view and also passes data to the view. It’s responsible for interpretation of the data sent by the view and dispersing that data to the appropriate models awaiting results to pass back to the view. Very little, if any business logic should be occurring in the controller.

The model’s job is to handle specific tasks related to a specific area of the application or functionality. Models will communicate directly with your database or other storage system and will handle business logic related to the results.

The view is passed data by the controller and is displayed to the user.

Overall, this question is worth knowing as the MVC design pattern has been used a lot in the last few years and is a very good design pattern to know. Even with more advanced flows that go down to repositories and entities, they still are following the same basic idea for the Controller and View. The Model is typically just split out into multiple components to handle specific tasks related to database data, business logic etc. The MVC design pattern helps draw a better understanding of what is being used, as a whole, in the industry.

Merlyn Coslett has over 15 years of experience developing applications and APIs using PHP. He also has had the pleasure of interviewing and hiring many Junior to Senior level developers over the last 3 years. Schedule a mock interview with Merlyn!

PHP Interview Question #19

How does one prevent the following Warning ‘Warning: Cannot modify header information – headers already sent’ and why does it occur in the first place?

(Question provided by Kordian Bruck)

Do not output anything to the browser before using code that modifies the HTTP headers. Once you call echo or any other code that clears the buffer you can no longer set cookies or headers. That is also true for error messages, so if an error happens before you use the header command and the INI directive display\_errors is set then that will also cause that error to show.

PHP Interview Question #20

What are SQL Injections, how do you prevent them and what are the best practices?

(Question provided by Kordian Bruck)

SQL injections are a method to alter a query in a SQL statement send to the database server. That modified query then might leak information like username/password combinations and can help the intruder to further compromise the server.

To prevent SQL injections, one should always check & escape all user input. In PHP, this is easily forgotten due to the easy access to $\_GET & $\_POST, and is often forgotten by inexperienced developers. But there are also many other ways that users can manipulate variables used in a SQL query through cookies or even uploaded files (filenames). The only real protection is to use prepared statements everywhere consistently.

Do not use any of the mysql\_\* functions which have been deprecated since PHP 5.5 ,but rather use PDO, as it allows you to use other servers than MySQL out of the box. mysqli\_\* are still an option, but there is no real reason nowadays not to use PDO, ODBC or DBA to get real abstraction. Ideally you want to use Doctrine or Propel to get rid of writing SQL queries all together and use object-relational mapping which binds your rows from the database to objects in your application.

PHP Interview Question #21

What does the following code output?

$i = 016;

echo $i / 2;

(Question provided by Kordian Bruck)

The Output should be 7. The leading zero indicates an octal number in PHP, so the number evaluates to the decimal number 14 instead to decimal 16.

PHP Interview Question #22

Why would you use === instead of ==?

(Question provided by Kordian Bruck)

If you would want to check for a certain type, like an integer or boolean, the === will do that exactly like one would expect from a strongly typed language, while == would convert the data temporarily and try to match both operand’s types. The identity operator (===) also performs better as a result of not having to deal with type conversion. Especially when checking variables for true/false you want to avoid using == as this would also take into account 0/1 or other similar representation.

Kordian Bruck is a freelance developer and consultant for over five years doing work in many areas of coding. Security and privacy in development are his fields of interest. Schedule a mock interview with Kordian!

PHP Interview Question #23

What are PSRs? Choose 1 and briefly describe it.

(Question provided by Jad Joubran)

PSRs are PHP Standards Recommendations that aim at standardising common aspects of PHP Development.

An example of a PSR is PSR-2, which is a coding style guide. More info on PSR-2 here.

Jad Joubran is the founder & CTO of eTobb.com. He is also an international Conference speaker Schedule a mock interview with Jad!

PHP Interview Question #24

What PSR Standards do you follow? Why would you follow a PSR standard?

(Question provided by Shawn Mayzes)

You should folow a PSR because coding standards often vary between developers and companies. This can cause issues when reviewing or fixing another developer’s code and finding a code structure that is different from yours. A PSR standard can help streamline the expectations of how the code should look, thus cutting down confusion and in some cases syntax errors.

PHP Interview Question #25

Do you use Composer? If yes, what benefits have you found in it?

(Question provided by Shawn Mayzes)

A: Using Composer is a tool for dependency management. You are able to declare the libraries your product relies on and Composer will manage the installation and updating of the libraries. The benefit is a consistent way of managing the libraries you depend on and you will spend less time managing the libraries you depend on in your project.

Shawn Mayzes has over 10 years of experience with PHP and is the CTO of Red Stamp Agency. He is also a former EA and DDB Software Engineer, and founded larachat.co, the Slack community for the PHP Framework Laravel Schedule a mock interview with Shawn!

Extra PHP Interview Practice Questions

Special thanks to Codementor Ben Edmunds for the following extra questions! Ben is the co-host of the PHP Town Hall podcast, author of Building Secure PHP Apps, and CTO at Mindfulware. When interviewing candidates, Ben tends to ask broad questions without exact answers to gauge both existing knowledge and the way someone thinks through a problem.

What’s the difference between using mysql\_ functions and PDO?