Top 50 PHP Interview Questions You Must Prepare in 2023

Q1. What are the common uses of PHP?

Uses of PHP
▪ It performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
▪ It can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
▪ You can add, delete, modify elements within your database with the help of PHP.
▪ Access cookies variables and set cookies.
▪ Using PHP, you can restrict users to access some pages of your website and also encrypt data.

Q2. What is PEAR in PHP?

PEAR is a **framework** and repository for reusable PHP components. PEAR stands for **PHP Extension and Application Repository**. It contains all types of PHP code snippets and libraries. It also provides a command line interface to install "packages" automatically.

Q3. What is the difference between static and dynamic websites?

Static Websites	Dynamic Websites
In static websites, content can't be changed after running the script. You cannot change anything in the site as it is predefined.	In dynamic websites, content of script can be changed at the run time. Its content is regenerated every time a user visits or reloads.

Q4. How to execute a PHP script from the command line?

To execute a PHP script, use the **PHP Command Line Interface (CLI)** and specify the file name of the script in the following way:

```
1 php script.php
```

Q5. Is PHP a case sensitive language?

PHP is **partially** case sensitive. The variable names are case-sensitive but function names are not. If you define the function name in lowercase and call them in uppercase, it will still work. User-defined functions are not case sensitive but the rest of the language is case-sensitive.

Q6. What is the meaning of 'escaping to PHP'?

The PHP parsing engine needs a way to **differentiate** PHP code from other elements in the page. The mechanism for doing so is known as 'escaping to PHP'. Escaping a string means to reduce **ambiguity** in quotes used in that string.

Q7. What are the characteristics of PHP variables?

Some of the important characteristics of PHP variables include:

- All variables in PHP are denoted with a leading **dollar sign** ($).
- The value of a variable is the value of its most recent assignment.
- Variables are **assigned** with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be **declared** before assignment.
- Variables in PHP do not have **intrinsic types** – a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have **default** values.

Q8. What are the different types of PHP variables?

There are 8 **data types** in PHP which are used to construct the variables:

1. **Integers** – are whole numbers, without a decimal point, like 4195.
2. **Doubles** – are floating-point numbers, like 3.14159 or 49.1.
3. **Booleans** – have only two possible values either true or false.
4. **NULL** – is a special type that only has one value: NULL.
5. **Strings** – are sequences of characters, like 'PHP supports string operations.'
6. **Arrays** – are named and indexed collections of other values.
7. **Objects** – are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
8. **Resources** – are special variables that hold references to resources external to PHP.

Q9. What are the rules for naming a PHP variable?

The following rules are needed to be followed while naming a **PHP variable**:

- Variable names must begin with a **letter** or **underscore** character.
- A variable name can consist of numbers, letters, underscores but you cannot use **characters** like + , − , % , (,) . & , etc.

Q10. What are the rules to determine the "truth" of any value which is not already of the Boolean type?

The rules to determine the "truth" of any value which is not already of the **Boolean** type are:

- If the value is a number, it is **false** if exactly equal to **zero** and true otherwise.
- If the value is a string, it is false if the string is **empty** (has zero characters) or is the string "0", and is true otherwise.
- Values of type **NULL** are always false.
- If the value is an **array**, it is false if it contains no other values, and it is true otherwise. For an object, containing a value means having a member variable that has been assigned a value.
- Valid resources are true (although some functions that return resources when they are successful will return FALSE when unsuccessful).
- Don't use **double** as Booleans.

Q11. What is NULL?

NULL is a special data type which can have only **one value**. A variable of data type NULL is a variable that has no value assigned to it. It can be assigned as follows:

```
1 $var = NULL;
```

The special constant NULL is capitalized by convention but actually it is case insensitive. So,you can also write it as :

```
1 $var = null;
```

A variable that has been assigned the NULL value, consists of the following **properties**:

- It evaluates to FALSE in a **Boolean** context.
- It returns FALSE when tested with **IsSet()** function.

Q12. How do you define a constant in PHP?

To define a constant you have to use **define()** function and to retrieve the value of a constant, you have to simply specifying its name.If you have defined a constant, it can never be changed or undefined. There is no need to have a constant with a $. A valid constant name starts with a letter or underscore.

Q13. What is the purpose of constant() function?

The constant() function will return the value of the constant. This is useful when you want to **retrieve** value of a constant, but you do not know its name, i.e., it is stored in a variable or returned by a function. For example –

```
1 <?php define("MINSIZE", 50); echo MINSIZE; echo constant("MINSIZE"); // same thing as the previous line ?>
```

Q14. What are the differences between PHP constants and variables?

Constants	Variables
There is no need to write dollar ($) sign before a constant	A variable must be written with the dollar ($) sign
Constants can only be defined using the define() function	Variables can be defined by simple assignment
Constants may be defined and accessed anywhere without regard to variable scoping rules.	In PHP, functions by default can only create and access variables within its own scope.
Constants cannot be redefined or undefined.	Variables can be redefined for each path individually.

Q15. Name some of the constants in PHP and their purpose.

1. **_LINE_** – It represents the current line number of the file.
2. **_FILE_** – It represents the full path and filename of the file. If used inside an include,the name of the included file is returned.
3. **_FUNCTION_** – It represents the function name.
4. **_CLASS_** – It returns the class name as it was declared.
5. **_METHOD_** – It represents the class method name.

Q16. What is the purpose of break and continue statement?

Break – It terminates the **for loop** or **switch** statement and transfers execution to the statement immediately following the for loop or switch.

Continue – It causes the **loop** to skip the remainder of its body and immediately retest its condition prior to reiterating.

Q17. What are the two most common ways to start and finish a PHP block of code?

The two most common ways to **start** and **finish** a PHP block of code are:

```
1 <?php [ --- PHP code---- ] ?>
```

```
1 <? [--- PHP code ---] ?>
```

Q18. What is the difference between PHP4 and PHP5?

PHP4	PHP5
• Constructor have same name as the Class name.	• Constructors are named as _construct and Destructors as _destruct().
• Everything is passed by value.	• All objects are passed by references.
• PHP4 does not declare a class as abstract	• PHP5 allows to declare a class as abstract
• It doesn't have static methods and properties in a class	• It allows to have static Methods and Properties in a class

Q19. What is the meaning of a final class and a final method?

The **final** keyword in a method declaration indicates that the method cannot be overridden by subclasses. A **class** that is declared **final** cannot be subclassed. This is particularly useful when we are creating an immutable class like the String class.Properties cannot be declared final, only **classes** and **methods** may be declared as final.

Q20. How can you compare objects in PHP?

We use the operator '==' to test if two objects are **instanced** from the same class and have same attributes and equal values. We can also test if two objects are referring to the same instance of the same class by the use of the identity operator '==='.

Q21. How can PHP and Javascript interact?

PHP and Javascript cannot directly interact since **PHP** is a **server side** language and **Javascript** is a **client-side** language. However, we can exchange variables since PHP can generate Javascript code to be executed by the browser and it is possible to pass specific variables back to PHP via the **URL.**

Q22. How can PHP and HTML interact?

It is possible to generate HTML through PHP scripts, and it is possible to pass pieces of **information** from **HTML** to PHP. PHP is a server side language and HTML is a client side language so PHP executes on server side and gets its results as strings, arrays, objects and then we use them to display its values in HTML.

Q23. Name some of the popular frameworks in PHP.

Some of the popular **frameworks** in PHP are:

- **CakePHP**
- **CodeIgniter**
- **Yii 2**
- **Symfony**
- **Zend Framework**

Q24. What are the data types in PHP?

PHP support 9 **primitive data types**:

Scalar Types	Compound Types	Special Types
- Integer - Boolean - Float - String	- Array - Object - Callable	- Resource - Null

Q25. What are constructor and destructor in PHP?

PHP constructor and destructor are special type functions which are automatically called when a PHP **class object** is **created** and **destroyed**. The constructor is the most useful of the two because it allows you to send parameters along when creating a new object, which can then be used to initialize variables on the object.

Here is an example of constructor and destructor in PHP:

```php
1
2  <?php class Foo { private $name; private $link; public function __construct($name) { $this->;name = $name;
3  }
4
5  public function setLink(Foo $link){
6  $this->;link = $link;
7  }
8
9  public function __destruct() {
   echo 'Destroying: ', $this->name, PHP_EOL;
10 }
11 }
12 ?>
```

Q26. What are include() and require() functions?

The **Include()** function is used to put data of one PHP file into another PHP file. If errors occur then the include() function produces a warning but does not **stop** the execution of the script and it will continue to execute.

The **Require()** function is also used to put data of one PHP file to another PHP file. If there are any errors then the require() function produces a **warning** and a fatal error and stops the execution of the script.

Q27. What is the main difference between require() and require_once()?

The require() includes and evaluates a specific file, while require_once() does that only if it has not been included before. The **require_once()** statement can be used to include a **php** file in another one, when you may need to include the called file more than once. So, require_once() is recommended to use when you want to include a file where you have a lot of functions.

Q28. What are different types of errors available in Php ?

The different types of **error** in PHP are:

- **E_ERROR**– A fatal error that causes script termination.
- **E_WARNING**– Run-time warning that does not cause script termination.
- **E_PARSE**– Compile time parse error.
- **E_NOTICE**– Run time notice caused due to error in code.
- **E_CORE_ERROR**– Fatal errors that occur during PHP initial startup.
- **E_CORE_WARNING**– Warnings that occur during PHP initial startup.
- **E_COMPILE_ERROR**– Fatal compile-time errors indication problem with script.
- **E_USER_ERROR**– User-generated error message.
- **E_USER_WARNING**– User-generated warning message.
- **E_USER_NOTICE**- User-generated notice message.
- **E_STRICT**– Run-time notices.
- **E_RECOVERABLE_ERROR**– Catchable fatal error indicating a dangerous error
- **E_ALL**– Catches all errors and warnings.

Q29. Explain the syntax for 'foreach' loop with example.

The foreach statement is used to **loop** through **arrays**. For each pass the value of the current array element is assigned to $value and the array pointer is moved by one and in the next pass next element will be processed.

Syntax-

foreach (array as value)
{
code to be executed;
}

Example-

```
<?php
1 $colors = array("blue", "white", "black");
2
3 foreach ($colors as $value) {
4 echo "$value
5 ";
```

```
6}
7?>
8
```

Q30. What are the different types of Array in PHP?

There are 3 **types of Arrays** in PHP:

1. **Indexed Array** – An array with a numeric index is known as the indexed array. Values are stored and accessed in linear fashion.
2. **Associative Array** – An array with strings as index is known as the associative array. This stores element values in association with key values rather than in a strict linear index order.
3. **Multidimensional Array** – An array containing one or more arrays is known as multidimensional array. The values are accessed using multiple indices.

Q31. What is the difference between single quoted string and double quoted string?

Singly quoted strings are treated almost literally, whereas doubly quoted strings replace variables with their values as well as specially interpreting certain character sequences. For example –

```
1
2 <?php
3 $variable = "name";
4 $statement = 'My $variable will not print!n';
5 print($statement);
6 print "
."
7 '
  $statement = "My $variable will print!n"
8 print($statement);
9 ?>
```
It will give the following **output**–

My $variable will not print!

My name will print

Q32. How to concatenate two strings in PHP?

To concatenate two string variables together, we use the **dot (.)** operator.

```
1 <?php $string1="Hello edureka"; $string2="123"; echo $string1 . " " . $string2; ?>
```
This will produce following **result** –

Hello edureka 123

Q33. How is it possible to set an infinite execution time for PHP script?

The set_time_limit(0) added at the beginning of a script sets to **infinite** the time of execution to not have the PHP error 'maximum execution time exceeded.' It is also possible to specify this in the php.ini file.

Q34. What is the difference between "echo" and "print" in PHP?

- PHP **echo** output one or more string. It is a language construct not a function. So use of parentheses is not required. But if you want to pass more than one parameter to echo, use of parentheses is required. Whereas, PHP **print** output a string. It is a language construct not a function. So use of parentheses is not required with the argument list. Unlike echo, it always returns 1.

- **Echo** can output one or more string but **print** can only output one string and always returns 1.
- **Echo** is faster than print because it does not return any value.

Q35. Name some of the functions in PHP.

Some of the functions in PHP include:

- **ereg()** – The ereg() function searches a string specified by string for a string specified by pattern, returning true if the pattern is found, and false otherwise.
- **ereg()** – The ereg() function searches a string specified by string for a string specified by pattern, returning true if the pattern is found, and false otherwise.
- **split()** – The split() function will divide a string into various elements, the boundaries of each element based on the occurrence of pattern in string.
- **preg_match()** – The preg_match() function searches string for pattern, returning true if pattern exists, and false otherwise.
- **preg_split()** – The preg_split() function operates exactly like split(), except that regular expressions are accepted as input parameters for pattern.

These were some of the most commonly asked basic level PHP interview questions. Let's move on to the next section of advanced level PHP interview questions.

Advanced level PHP Interview Questions

Q36. What is the main difference between asp net and PHP?

PHP is a **programming language** whereas **ASP.NET** is a programming **framework**. Websites developed by ASP.NET may use C#, but also other languages such as J#. ASP.NET is compiled whereas PHP is interpreted. ASP.NET is designed for windows machines, whereas PHP is platform free and typically runs on Linux servers.

Q37. What is the use of session and cookies in PHP?

A session is a global variable stored on the server. Each session is assigned a unique id which is used to retrieve stored values. Sessions have the capacity to store relatively large data compared to cookies. The session values are automatically deleted when the browser is closed.

Q38. What is overloading and overriding in PHP?

Overloading is defining functions that have **similar signatures**, yet have **different parameters**. Overriding is only pertinent to derived classes, where the parent class has defined a **method** and the derived class wishes to **override** that method. In PHP, you can only overload methods using the magic method __call.

Q40. What is the difference between $message and $$message in PHP?

They are both variables. But $message is a variable with a **fixed name**. $$message is a variable whose name is **stored** in $message. For example, if $message contains "var", $$message is the same as $var.

Q41. How can we create a database using PHP and MySQL?

The basic steps to create MySQL database using PHP are:

- Establish a **connection** to MySQL server from your PHP script.
- If the connection is successful, write a SQL query to create a **database** and store it in a string variable.
- **Execute** the query.

Q42. What is GET and POST method in PHP?

The GET method sends the **encoded** user information appended to the page request. The page and the encoded information are separated by the **? character**. For example –

```
1 <a href="http://www.test.com/index.htm?name1=value1&name2=value2">http://www.test.com/index.htm?nam
```

The POST method transfers information via **HTTP** headers. The information is encoded as described in case of GET method and put into a header called **QUERY_STRING**.

Q43. What is the difference between GET and POST method?

GET	POST
• The GET method is restricted to send upto 1024 characters only.	• The POST method does not have any restriction on data size to be sent.
• GET can't be used to send binary data, like images or word documents, to the server.	• The POST method can be used to send ASCII as well as binary data.
• The data sent by GET method can be accessed using QUERY_STRING environment variable.	• The data sent by POST method goes through HTTP header so security depends on HTTP protocol.
• The PHP provides $_GET associative array to access all the sent information using GET method.	• The PHP provides $_POST associative array to access all the sent information using POST method.

Q44. What is the use of callback in PHP?

PHP callback are functions that may be called **dynamically** by PHP. They are used by native functions such as **array_map, usort, preg_replace_callback**, etc. A callback function is a function that you create yourself, then pass to another function as an argument. Once it has access to your callback function, the receiving function can then call it whenever it needs to.

Here is a basic example of callback function –

```php
1 <?php
2
3 function thisFuncTakesACallback($callbackFunc)
4 {
5 echo "I'm going to call $callbackFunc!
6 ";
7 $callbackFunc();
8 }
9 }
10 function thisFuncGetsCalled()
11 {
```

```
12  echo "I'm a callback function!
13  ";
14  }
15
16  thisFuncTakesACallback( 'thisFuncGetsCalled' );

    ?>
```

Q45. What is a lambda function in PHP?

A lambda function is an **anonymous** PHP function that can be stored in a variable and passed as an argument
to other functions or methods. A closure is a lambda function that is aware of its surrounding context. For
example –

```
1  $input = array(1, 2, 3, 4, 5);
2  $output = array_filter($input, function ($v) { return $v > 2; });
```

function ($v) { return $v > 2; } is the lambda function definition. We can store it in a variable so that it can
be reusable.

Q46. What are PHP Magic Methods/Functions?

In PHP all functions starting with __ names are magical functions/methods. These **methods**, identified by a
two underscore prefix (__), **function** as interceptors that are automatically called when certain conditions
are met. **PHP** provides a number of '**magic**' **methods** that allow you to do some pretty neat tricks in object
oriented programming.

Here are list of **Magic Functions** available in PHP

__destruct()	__sleep()
__construct()	__wakeup()
__call()	__toString()
__get()	__invoke()
__set()	__set_state()
__isset()	__clone()
__unset()	__debugInfo()

Q47. How can you encrypt password using PHP?

The crypt () function is used to create one way **encryption**. It takes one input string and one optional
parameter. The function is defined as: crypt (input_string, salt), where input_string consists of the string
that has to be encrypted and salt is an optional parameter. PHP uses **DES** for encryption. The format is as
follows:

```
1  <?php $password = crypt('edureka'); print $password. "is the encrypted version of edureka"; ?>
```

Q48. How to connect to a URL in PHP?

PHP provides a library called **cURL** that may already be included in the installation of PHP by default. cURL stands for client URL, and it allows you to connect to a URL and retrieve information from that page such as the HTML content of the page, the HTTP headers and their associated data.

Q49. What is Type hinting in PHP?

Type hinting is used to specify the **expected data** type of an argument in a function declaration. When you call the function, PHP will check whether or not the arguments are of the specified type. If not, the run-time will raise an error and execution will be halted.

Here is an example of **type hinting**–

```
1 <?php function sendEmail (Email $email) { $email->send();
2 }
3 ?>
```

The example shows how to send Email function argument $email Type hinted of Email Class. It means to call this function you must have to pass an email object otherwise an error is generated.

Q50. What is the difference between runtime exception and compile time exception?

An exception that occurs at **compile time** is called a checked exception. This exception cannot be ignored and must be handled carefully. For example, if you use **FileReader** class to **read** data from the file and the file specified in class constructor does not exist, then a FileNotFoundException occurs and you will have to manage that exception. For the purpose, you will have to write the code in a try-catch block and handle the exception. On the other hand, an exception that occurs at runtime is called unchecked-exception.