**PHP SETUP=>**

**Xampp downloaded**

Path: "D:\PHP Workspace\xampp"

**Composer downloaded**

**Path:**

PHP version 8.2.12

D:\PHP Workspace\xampp\php\php.exe

Proxy: none

Add to System path:

D:\PHP Workspace\xampp\php

**Laravel downloaded**

C:\Users\Asus>composer global require laravel/installer

Changed current directory to C:/Users/Asus/AppData/Roaming/Composer

./composer.json has been created

Running composer update laravel/installer

……………………………………………

18 package suggestions were added by new dependencies, use `composer suggest` to see details.

Generating autoload files

17 packages you are using are looking for funding.

Use the `composer fund` command to find out more!

No security vulnerability advisories found.

Using version ^5.5 for laravel/installer

**To Do Programs:**

C:\Users\Asus>d:

D:\>cd PHP Workspace

D:\PHP Workspace>cd Laravel

D:\PHP Workspace>Laravel>laravel new projectname

After completing setup type

D:\PHP Workspace\Laravel>code .

It will open working directory in vs code after click **terminal>new** button

After that first go in project directory then, Type in terminal

Asus@ASUS-VIVOBOOK-PRO-15 MINGW64 /d/PHP Workspace/Laravel/pizzahouse

$ php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

|  |  |
| --- | --- |
| **PHP** | |
|  | |
|  | |
| [**Paradigm**](https://en.wikipedia.org/wiki/Programming_paradigm) | [Multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language): [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [procedural](https://en.wikipedia.org/wiki/Procedural_programming), [reflective](https://en.wikipedia.org/wiki/Reflective_programming) |
| [**Designed by**](https://en.wikipedia.org/wiki/Software_design) | [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) |
| [**Developer**](https://en.wikipedia.org/wiki/Software_developer) | [The PHP Development Team](https://php.net/credits/), [Zend Technologies](https://en.wikipedia.org/wiki/Zend_Technologies), [PHP Foundation](https://thephp.foundation/) |
| **First appeared** | 8 June 1995; 28 years ago[[1]](https://en.wikipedia.org/wiki/PHP#cite_note-history-php-release-1)[[2]](https://en.wikipedia.org/wiki/PHP#cite_note-mysqlconference-2) |
|  | |
| [**Stable release**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 8.3.3 / 15 February 2024; 9 days ago[[3]](https://en.wikipedia.org/wiki/PHP" \l "cite_note-3) |
| [**Typing discipline**](https://en.wikipedia.org/wiki/Type_system) | [Dynamic](https://en.wikipedia.org/wiki/Dynamic_typing), [weak](https://en.wikipedia.org/wiki/Strong_and_weak_typing), [gradual](https://en.wikipedia.org/wiki/Gradual_typing)[[4]](https://en.wikipedia.org/wiki/PHP#cite_note-4) |
| **Implementation language** | [C](https://en.wikipedia.org/wiki/C_(programming_language)) (primarily; some components [C++](https://en.wikipedia.org/wiki/C%2B%2B)) |
| [**OS**](https://en.wikipedia.org/wiki/Operating_system) | [Unix-like](https://en.wikipedia.org/wiki/Unix-like), [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [IBM i](https://en.wikipedia.org/wiki/IBM_i), [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS) |
| [**License**](https://en.wikipedia.org/wiki/Software_license) | dual licensed [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) version 2 or any later version and PHP License for PHP versions 3.0 or earlier.[[5]](https://en.wikipedia.org/wiki/PHP#cite_note-5) Only [PHP License](https://en.wikipedia.org/wiki/PHP_License) (most of Zend engine under [Zend Engine License](https://en.wikipedia.org/wiki/Zend_Engine_License)) for 3.01x and later versions. |
| [**Filename extensions**](https://en.wikipedia.org/wiki/Filename_extension) | .php,.phar,.phtml,.pht,.phps |
| **Website** | [www.php.net](https://www.php.net/) [Edit this at Wikidata](https://www.wikidata.org/wiki/Q59#P856) |
| **Major**[**implementations**](https://en.wikipedia.org/wiki/Programming_language_implementation) | |
| [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine), [HHVM](https://en.wikipedia.org/wiki/HHVM), [PeachPie](https://en.wikipedia.org/wiki/PeachPie" \o "PeachPie), [Quercus](https://en.wikipedia.org/wiki/Quercus_(software)), [Parrot](https://en.wikipedia.org/wiki/Parrot_virtual_machine) | |
| **Influenced by** | |
| [Perl](https://en.wikipedia.org/wiki/Perl), [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)),[[6]](https://en.wikipedia.org/wiki/PHP#cite_note-6) [Tcl](https://en.wikipedia.org/wiki/Tcl" \o "Tcl),[[2]](https://en.wikipedia.org/wiki/PHP#cite_note-mysqlconference-2) [JavaScript](https://en.wikipedia.org/wiki/JavaScript)[[7]](https://en.wikipedia.org/wiki/PHP#cite_note-optimisation_ideas-7) | |
| **Influenced** | |
| [Hack](https://en.wikipedia.org/wiki/Hack_(programming_language)), [JSP](https://en.wikipedia.org/wiki/Jakarta_Server_Pages), [ASP](https://en.wikipedia.org/wiki/Active_Server_Pages), [React JS](https://en.wikipedia.org/wiki/React_(software)) | |
| * [PHP Programming](https://en.wikibooks.org/wiki/PHP_Programming) at Wikibooks | |

# PHP Interview Questions

There is given PHP interview questions and answers that have been asked in many companies. Let's see the list of top PHP interview questions.

### 1) What is PHP?

**PHP** stands for Hypertext Preprocessor. It is an open-source **server-side scripting** language which is widely used for web development. It supports many databases like MySQL, Oracle, Sybase, Solid, PostgreSQL, generic ODBC etc.

[More Details...](https://www.javatpoint.com/what-is-php)

### 2) 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.

### 3) Who is known as the father of PHP?

Rasmus Lerdorf

### 4) What was the old name of PHP?

The old name of PHP was Personal Home Page.

### 5) Explain the difference b/w static and dynamic websites?

In **static websites**, content can't be changed after running the script. You can't change anything on the site. It is predefined.

In **dynamic websites**, content of script can be changed at the run time. Its content is regenerated every time a user visit or reload. Google, yahoo and every search engine is the example of dynamic website.

### 6) What is the name of scripting engine in PHP?

The scripting engine that powers PHP is called Zend Engine 2.

### 7) Explain the difference between PHP4 and PHP5.

PHP4 doesn't support oops concept and uses Zend Engine 1.

PHP5 supports oops concept and uses Zend Engine 2.

### 8) What are the popular Content Management Systems (CMS) in PHP?

* **WordPress:** WordPress is a free and open-source content management system (CMS) based on PHP & MySQL. It includes a plug-in architecture and template system. It is mostly connected with blogging but supports another kind of web content, containing more traditional mailing lists and forums, media displays, and online stores.
* **Joomla:** Joomla is a free and open-source content management system (CMS) for distributing web content, created by Open Source Matters, Inc. It is based on a model-view-controller web application framework that can be used independently of the CMS.
* **Magento:** Magento is an open source E-trade programming, made by Varien Inc., which is valuable for online business. It has a flexible measured design and is versatile with many control alternatives that are useful for clients. Magento utilizes E-trade stage which offers organization extreme E-business arrangements and extensive support network.
* **Drupal:** Drupal is a CMS platform developed in PHP and distributed under the GNU (General Public License).

### 9) What are the popular frameworks in PHP?

* CakePHP
* CodeIgniter
* Yii 2
* Symfony
* Zend Framework etc.

### 10) Which programming language does PHP resemble to?

PHP has borrowed its syntax from Perl and C.

### 11) List some of the features of PHP7.

* Scalar type declarations
* Return type declarations
* Null coalescing operator (??)
* Spaceship operator
* Constant arrays using define()
* Anonymous classes
* Closure::call method
* Group use declaration
* Generator return expressions
* Generator delegation
* Space ship operator

### 12) What is "echo" in PHP?

PHP echo output one or more string. It is a language construct not a function. So the use of parentheses is not required. But if you want to pass more than one parameter to echo, the use of parentheses is required.

**Syntax:**

1. void echo ( string $arg1 [, string $... ] )

[More details...](https://www.javatpoint.com/php-echo)

### 13) What is "print" in PHP?

PHP print output a string. It is a language construct not a function. So the use of parentheses is not required with the argument list. Unlike echo, it always returns 1.

**Syntax:**

1. int print ( string $arg)

[More details...](https://www.javatpoint.com/php-print)

### 14) What is the difference between "echo" and "print" in PHP?

**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.

### 15) How a variable is declared in PHP?

A PHP variable is the name of the memory location that holds data. It is temporary storage.

**Syntax:**

1. $variableName=value;

[More details...](https://www.javatpoint.com/php-variables)

### 16) What is the difference between $message and $$message?

**$message** stores variable data while **$$message** is used to store variable of variables.

$message stores fixed data whereas the data stored in $$message may be changed dynamically.

[More Details...](https://www.javatpoint.com/php-dollar-doubledollar)

### 17) What are the ways to define a constant in PHP?

PHP constants are name or identifier that can't be changed during execution of the script. PHP constants are defined in two ways:

* Using define() function
* Using const() function

[More details...](https://www.javatpoint.com/php-constants)

### 18) What are magic constants in PHP?

PHP magic constants are predefined constants, which change based on their use. They start with a double underscore (\_\_) and end with a double underscore (\_\_).

[More Details...](https://www.javatpoint.com/php-magic-constants)

### 19) How many data types are there in PHP?

PHP data types are used to hold different types of data or values. There are 8 primitive data types which are further categorized in 3 types:

* Scalar types
* Compound types
* Special types

[More Details...](https://www.javatpoint.com/php-data-types)

### 20) How to do single and multi line comment in PHP?

PHP single line comment is made in two ways:

* Using // (C++ style single line comment)
* Using # (Unix Shell style single line comment)

PHP multi-line comment is made by enclosing all lines within.

[More details...](https://www.javatpoint.com/php-comments)

### 21) What are the different loops in PHP?

For, while, do-while and for each.

### 22) What is the use of count() function in PHP?

The PHP count() function is used to count total elements in the array, or something an object.

### 23) What is the use of header() function in PHP?

The header() function is used to send a raw HTTP header to a client. It must be called before sending the actual output. For example, you can't print any HTML element before using this function.

### 24) What does isset() function?

The isset() function checks if the variable is defined and not null.

### 25) Explain PHP parameterized functions.

PHP parameterized functions are functions with parameters. You can pass any number of parameters inside a function. These given parameters act as variables inside your function. They are specified inside the parentheses, after the function name. Output depends upon dynamic values passed as parameters into the function.

[More details...](https://www.javatpoint.com/php-parameterized-function)

### 26) Explain PHP variable length argument function

PHP supports variable length argument function. It means you can pass 0, 1 or n number of arguments in function. To do this, you need to use 3 ellipses (dots) before the argument name. The 3 dot concept is implemented for variable length argument since PHP 5.6.

[More details...](https://www.javatpoint.com/php-variable-length-argument-function)

### 27) Explain PHP variable length argument function.

PHP supports variable length argument function. It means you can pass 0, 1 or n number of arguments.

### 28) What is the array in PHP?

An array is used to store multiple values in a single value. In PHP, it orders maps of pairs of keys and values. It saves the collection of the data type.

[More Details...](https://www.javatpoint.com/php-array)

### 29) How many types of array are there in PHP?

There are three types of array in PHP:

1. **Indexed array:** an array with a numeric key.
2. **Associative array:** an array where each key has its specific value.
3. **Multidimensional array:** an array containing one or more arrays within itself.

### 30) Explain some of the PHP array functions?

There are many array functions in PHP:

* array()
* array\_change\_key\_case()
* array\_chunk()
* count()
* sort()
* array\_reverse()
* array\_search()
* array\_intersect()

[More details...](https://www.javatpoint.com/php-array-functions)

### 31) What is the difference between indexed and associative array?

The indexed array holds elements in an indexed form which is represented by number starting from 0 and incremented by 1. For example:

1. $season=**array**("summer","winter","spring","autumn");

The associative array holds elements with name. For example:

1. $salary=**array**("Sonoo"=>"350000","John"=>"450000","Kartik"=>"200000");

[More Details...](https://www.javatpoint.com/php-array)

### 32) How to get the length of string?

The strlen() function is used to get the length of the string.

[More Details...](https://www.javatpoint.com/php-string-functions)

### 33) Explain some of the PHP string functions?

There are many array functions in PHP:

* strtolower()
* strtoupper()
* ucfirst()
* lcfirst()
* ucwords()
* strrev()
* strlen()

[More details...](https://www.javatpoint.com/php-string-functions)

### 34) What are the methods to submit form in PHP?

There are two methods GET and POST.

[More Details...](https://www.javatpoint.com/php-form)

### 35) How can you submit a form without a submit button?

You can use JavaScript submit() function to submit the form without explicitly clicking any submit button.

### 36) What are the ways to include file in PHP?

PHP allows you to include file so that page content can be reused again. There are two ways to add the file in PHP.

1. include
2. require

[More details...](https://www.javatpoint.com/php-include-file)

### 37) Differentiate between require and include?

Require and include both are used to include a file, but if data is not found include sends warning whereas require sends Fatal error.

[More Details...](https://www.javatpoint.com/php-include-file)

### 38) Explain setcookie() function in PHP?

PHP setcookie() function is used to set cookie with HTTP response. Once the cookie is set, you can access it by $\_COOKIE superglobal variable.

**Syntax:**

1. bool setcookie ( string $name [, string $value [, int $expire = 0 [, string $path
2. [, string $domain [, bool $secure = false [, bool $httponly = false ]]]]]] )

[More details...](https://www.javatpoint.com/php-cookie)

### 39) How can you retrieve a cookie value?

1. echo $\_COOKIE ["user"];

[More Details...](https://www.javatpoint.com/php-cookie)

### 40) What is a session?

PHP Engine creates a logical object to preserve data across subsequent HTTP requests, which is known as session.

Sessions generally store temporary data to allow multiple PHP pages to offer a complete functional transaction for the same user.

Simply, it maintains data of an user (browser).

[More Details...](https://www.javatpoint.com/php-session)

### 41) What is the method to register a variable into a session?

1. <?php
2. Session\_register($ur\_session\_var);
3. ?>

### 42) What is $\_SESSION in PHP?

A session creates a file in a temporary directory on the server where registered session variables and their session id are stored. This data will be available to all pages on the site amid that visit.

The area of the temporary record is controlled by a setting in the php.ini document called session.save\_path.

At the point when a session is begun following things happen -

1. PHP first makes two duplicates of one of a kind session id for that particular session of the client which is an arbitrary string of 32 hexadecimal numbers, for example, 3c7foj34c3jjhkyepop2fc937e3443.
2. One copy of unique session id automatically sent to the user?s computer for the sake of synchronization in future ahead, and one copy is being maintained at server side till the session is running.
3. Whenever you want to access the page of website or web app, then session id of the current user will be associated with the HTTP header, and that will be compared by the session id which is being maintained at the server. After completing the comparison process, you can easily access the page of the website or web app
4. A session ends when the user closes the browser, or after leaving the site, the server will terminate the session after a predetermined period, commonly 30 minutes duration.

### 43) What is PHP session\_start() and session\_destroy() function?

PHP session\_start() function is used to start the session. It starts new or resumes the current session. It returns the current session if the session is created already. If the session is not available, it creates and returns new sessions.

[More details...](https://www.javatpoint.com/php-session)

### 44) What is the difference between session and cookie?

The main difference between session and cookie is that cookies are stored on user's computer in the text file format while sessions are stored on the server side.

Cookies can't hold multiple variables, on the other hand, Session can hold multiple variables.

You can manually set an expiry for a cookie, while session only remains active as long as browser is open.

### 45) Write syntax to open a file in PHP?

PHP fopen() function is used to open file or URL and returns resource. It accepts two arguments: $filename and $mode.

**Syntax:**

1. resource fopen ( string $filename , string $mode [, bool $use\_include\_path = false [, resource $context ]] )

[More details...](https://www.javatpoint.com/php-open-file)

### 46) How to read a file in PHP?

PHP provides various functions to read data from the file. Different functions allow you to read all file data, read data line by line, and read data character by character.

PHP file read functions are given below:

* fread()
* fgets()
* fgetc()

[More details...](https://www.javatpoint.com/php-read-file)

### 47) How to write in a file in PHP?

PHP fwrite() and fputs() functions are used to write data into file. To write data into a file, you need to use w, r+, w+, x, x+, c or c+ mode.

[More details...](https://www.javatpoint.com/php-write-file)

### 48) How to delete file in PHP?

The unlink() function is used to delete a file in PHP.

1. bool unlink (string $filename)

[More Details...](https://www.javatpoint.com/php-delete-file)

### 49) What is the method to execute a PHP script from the command line?

You should just run the PHP command line interface (CLI) and specify the file name of the script to be executed as follows.

### 50) How to upload file in PHP?

The move\_uploaded\_file() function is used to upload file in PHP.

1. bool move\_uploaded\_file ( string $filename , string $destination )

[More Details...](https://www.javatpoint.com/php-file-upload)

### 51) How to download file in PHP?

The readfile() function is used to download the file in PHP.

1. int readfile ( string $filename )

[More Details...](https://www.javatpoint.com/php-download-file)

### 52) How can you send email in PHP?

The mail() function is used to send email in PHP.

1. bool mail($to,$subject,$message,$header);

[More Details...](https://www.javatpoint.com/php-mail)

### 53) How do you connect MySQL database with PHP?

There are two methods to connect MySQL database with PHP. Procedural and object-oriented style.

[More Details...](https://www.javatpoint.com/php-mysql-connect)

### 54) How to create connection in PHP?

The mysqli\_connect() function is used to create a connection in PHP.

1. resource mysqli\_connect (server, username, password)

[More Details...](https://www.javatpoint.com/php-mysql-connect)

### 55) How to create database connection and query in PHP?

Since PHP 4.3, mysql\_reate\_db() is deprecated. Now you can use the following 2 alternatives.

* mysqli\_query()
* PDO::\_query()

[More details...](https://www.javatpoint.com/php-mysql-create)

### 56) How can we increase execution time of a PHP script?

By default, the maximum execution time for PHP scripts is set to 30 seconds. If a script takes more than 30 seconds, PHP stops the script and returns an error.

You can change the script run time by changing the max\_execution\_time directive in the php.ini file.

When a script is called, set\_time\_limit function restarts the timeout counter from zero. It means, if default timer is set to 30 sec, and 20 sec is specified in function set\_time\_limit(), then script will run for 45 seconds. If 0sec is specified in this function, script takes unlimited time.

### 57) What are the different types of errors in PHP?

There are 3 types of error in PHP.

1. **Notices:**These are non-critical errors. These errors are not displayed to the users.
2. **Warnings:**These are more serious errors, but they do not result in script termination. By default, these errors are displayed to the user.
3. **Fatal Errors:**These are the most critical errors. These errors may cause due to immediate termination of script.

### 58) How to stop the execution of PHP script?

The exit() function is used to stop the execution of PHP script.

### 59) What are the encryption functions in PHP?

CRYPT() and MD5()

### 60) What is htaccess in PHP?

The .htaccess is a configuration file on Apache server. You can change configuration settings using directives in Apache configuration files like .htaccess and httpd.conf.

### 61) Explain PHP explode() function.

The PHP explode() function breaks a string into an array.

### 62) Explain PHP split() function.

The PHP split() function splits string into an array by regular expression.

### 63) How can we get IP address of a client in PHP?

1. $\_SERVER["REMOTE\_ADDR"];

### 64) What is the meaning of a Persistent Cookie?

A persistent cookie is permanently stored in a cookie file on the browser's computer. By default, cookies are temporary and are erased if we close the browser.

### 65) What is the use of the function 'imagetypes()'?

imagetypes() gives the image format and types supported by the current version of GD-PHP.

### 66) 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 script.

### 67) What is Cookies? How to create cookies in PHP?

A cookie is used to identify a user. A cookie is a little record that the server installs on the client's Computer. Each time a similar PC asks for a page with a program, it will send the cookie as well. With PHP, you can both make and recover cookie value.

**Some important points regarding Cookies:**

1. Cookies maintain the session id generated at the back end after verifying the user's identity in encrypted form, and it must reside in the browser of the machine
2. You can store only string values not object because you can't access any object across the website or web apps
3. Scope: - Multiple pages.
4. By default, cookies are temporary and transitory cookie saves in the browser only.
5. By default, cookies are URL particular means Gmail isn't supported in Yahoo and the vice versa.
6. Per site 20 cookies can be created in one website or web app
7. The Initial size of the cookie is 50 bytes.
8. The Maximum size of the cookie is 4096 bytes.

### 68) What is the Importance of Parser in PHP?

PHP parser parses the PHP developed website from the opening to the closing tag. Tags indicate that from where PHP code is being started and ended. In other words, opening and closing tags decide the scope of PHP scripting syntax of closing tag in PHP

<?php syntax of opening tag in PHP  
?> syntax of closing tag in PHP

### 69) How can we create a database using PHP and MySQL?

The necessary steps to create a 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.

|  |  |
| --- | --- |
| [**Developer(s)**](https://en.wikipedia.org/wiki/Programmer) | Laravel logo  Taylor Otwell |
| **Initial release** | June 2011; 12 years ago[[1]](https://en.wikipedia.org/wiki/Laravel" \l "cite_note-maxoffsky-1) |
| [**Stable release**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 10.3.3[[2]](https://en.wikipedia.org/wiki/Laravel#cite_note-wikidata-40dd29aa83498413da05a1ea64952745c1633fb5-v10-2) [Edit this on Wikidata](https://www.wikidata.org/wiki/Q13634357?uselang=en#P348) / 13 February 2024; 10 days ago |
| [**Repository**](https://en.wikipedia.org/wiki/Repository_(version_control)) | [github.com/laravel/framework](https://github.com/laravel/framework) |
| **Written in** | [PHP](https://en.wikipedia.org/wiki/PHP) |
| [**Type**](https://en.wikipedia.org/wiki/Software_categories#Categorization_approaches) | [Web framework](https://en.wikipedia.org/wiki/Web_framework) |
| [**License**](https://en.wikipedia.org/wiki/Software_license) | [MIT License](https://en.wikipedia.org/wiki/MIT_License) |
| **Website** | [laravel.com](https://laravel.com/) |
|  |  |

Laravel Interview Questions

A list of top frequently asked **Laravel Interview Questions and answers** are given below.

1) What is Laravel?

Laravel is free to use, open-source web framework based on PHP. It is developed by **Taylor Otwell**. It supports the MVC (Model-View-Controller) architectural pattern. Laravel provides an expressive and elegant syntax, which is useful for creating a wonderful web application easily and quickly. The first version of Laravel was released on **9th June 2011**.

As of SitePoint survey in March 2015, Laravel was voted as one of the most popular PHP frameworks along with **Symfony, Nette, CodeIgniter, and Yii2**.

2) What are the main features of Laravel?

Some of the main features of Laravel are:

* Eloquent ORM
* Query builder
* Reverse Routing
* Restful Controllers
* Migrations
* Database Seeding
* Unit Testing
* Homestead

3) What do you understand by Eloquent ORM?

**Eloquent ORM (Object-Relational Mapping)** is one of the main features of the Laravel framework. It may be defined as an advanced PHP implementation of the active record pattern.

*Active record pattern is an architectural pattern which is found in software. It is responsible for keeping in-memory object data in relational databases*

Eloquent ORM is also responsible for providing the internal methods at the same time when enforcing constraints on the relationship between database objects. Eloquent ORM represents database tables as classes, with their object instances tied to single table rows, while following the active record pattern.

4) What is Query Builder in Laravel?

Laravel's Query Builder provides more direct access to the database, alternative to the Eloquent ORM. It doesn't require SQL queries to be written directly. Instead, it offers a set of classes and methods which are capable of building queries programmatically. It also allows specific caching of the results of the executed queries.

5) Write down the name of some aggregates methods provided by the Laravel's query builder.

Some of the methods that Query Builder provides are:

* count()
* max()
* min()
* avg()
* sum()

6) What is routing?

All Laravel routes are defined in route files, which are stored in the routes directory. These files are loaded by the MVC framework. The routes/web.php files define routes that are available for the web interface. Those routes are allotted as the web middleware group, which provide features such as **session state** and **CSRF** protection. The routes available in **routes/api.php** are stateless and are allotted as the API middleware group. For most of the applications, one should start by defining routes in routes/web.php file.

7) What do you understand by Reverse routing?

Reverse routing in Laravel is used to generate the URL based on name or symbol. It defines a relationship between the links and, Laravel routes, and it is possible to make later changes to the routes to be automatically propagated into relevant links. When the links are generated using names of existing routes, the appropriate uniform resource identifiers (URIs) are automatically generated by Laravel. Reverse routing provides flexibility to the application and helps the developer to write cleaner codes.

Route Declaration:

1. Route::get('login', 'users@login');

A link can be created to it using reverse routing, which can be further transferred in any parameter that we have defined. If optional parameters are not supplied, they are removed automatically from the generated links.

1. {{ HTML::link\_to\_action('users@login') }}

By using it, a URL like [https://abc.go.com/loginwill](http://abc.com/loginwill) be created automatically.

8) How will you describe Bundles in Laravel?

In Laravel, Bundles are also known as **Packages**. Packages are the primary way to add more functionality to Laravel. Packages can be anything, from a great way to work with dates like Carbon, or an entire BDD testing framework like **Behat**. Laravel also provides support for creating custom packages.

There are different types of packages. Some of them are stand-alone packages. This means they can work with any PHP framework. The frameworks like **Carbon and Behat** are examples of stand-alone packages. Other packages are intended for use with Laravel. These packages may contain routes, controllers, views, and configurations which are mainly designed to enhance a Laravel application.

9) What is a composer, and how can we install Laravel by the composer?

A composer is a dependency manager in PHP. It manages the dependencies which are required for a project. It means that the composer will pull in all the necessary libraries, dependencies, and manage all at a single place.

**Laravel Installation Steps:**

* If you don't have a composer on a system, download composer from <https://getcomposer.org/download/>
* Open command prompt
* Go to htdocs folder
* Run the below command under C:\xampp\htdocs>

10) Does Laravel support caching?

Yes, Laravel provides support for popular caching backends like **Memcached** and **Redis**.

By default, Laravel is configured to use file cache driver, which is used to store the serialized or cached objects in the file system. For huge projects, it is suggested to use Memcached or Redis.

11) How to clear cache in Laravel?

The syntax to clear cache in Laravel is given below:

* php artisan cache: clear
* php artisan config: clear
* php artisan cache: clear

12) How will you explain middleware in Laravel?

As the name suggests, middleware works as a middleman between request and response. Middleware is a form of HTTP requests filtering mechanism. For example, Laravel consists of middleware which verifies whether the user of the application is authenticated or not. If a user is authenticated and trying to access the dashboard then, the middleware will redirect that user to home page; otherwise, a user will be redirected to the login page.

There are two types of middleware available in Laravel:

**Global Middleware**

It will run on every HTTP request of the application.

**Route Middleware**

It will be assigned to a specific route.

**Syntax**

1. php artisan make:middlewareMiddelwareName

**Example**

1. php artisan make:middlewareUserMiddleware

Now, UserMiddleware.php file will be created in app/Http/Middleware.

13) What do you understand by database migrations in Laravel? How can we use it?

Migrations can be defined as version control for the database, which allows us to modify and share the application's database schema easily. Migrations are commonly paired with **Laravel's schema builder** to build the application's database schema easily.

A migration file includes two methods, **up()** and **down()**. A method up() is used to add new tables, columns or indexes database and the down() method is used to reverse the operations performed by the up() method.

We can generate a migration and its file by using the **make:migration**.

**Syntax**

1. php artisan make:migration blog

By using it, a current date blog.php file will be created in database/migrations.

14) What do you know about Service providers in Laravel?

Service providers can be defined as the central place to configure all the entire Laravel applications. Applications, as well as Laravel's core services, are bootstrapped via service providers. These are powerful tools for maintaining class dependencies and performing dependency injection. Service providers also instruct Laravel to bind various components into the Laravel's Service Container.

An artisan command is given here which can be used to generate a service provider:

1. php artisan make: provider ClientsServiceProvider

Almost, all the service providers extend the Illuminate\Support\ServiceProviderclass. Most of the service providers contain below-listed functions in its file:

* Register() Function
* Boot() Function

Within the Register() method, one should only bind things into the service container. One should never attempt to register any event listeners, routes, or any other piece of functionality within the Register() method.

15) How can we get data between two dates using Query in Laravel?

We can use **whereBetween()** method to retrieve the data between two dates with Query.

**Example**

1. Blog::whereBetween('created\_at', [$date1, $date2])->get();

16) What do you know about CSRF token in Laravel? How can someone turn off CSRF protection for a specific route?

CSRF protection stands for **Cross-Site Request Forgery** protection. CSRF detects unauthorized attacks on web applications by the unauthorized users of a system. The built-in CSRF plug-in is used to create CSRF tokens so that it can verify all the operations and requests sent by an active authenticated user.

To turn off CSRF protection for a specific route, we can add that specific URL or Route in $except variable which is present in the app\Http\Middleware\VerifyCsrfToken.phpfile.

**Example**

1. classVerifyCsrfToken **extends** BaseVerifier
2. {
3. **protected** $except = [
4. 'Pass here your URL',
5. ];
6. }

17) List some official packages provided by Laravel?

There are some official packages provided by Laravel which are given below:

**Cashier**

Laravel cashier implements an expressive, fluent interface to Stripe's and Braintree's subscription billing services. It controls almost all of the boilerplate subscription billing code you are dreading writing. Moreover, the cashier can also control coupons, subscription quantities, swapping subscription, cancellation grace periods, and even generate invoice PDFs.

**Envoy**

Laravel Envoy is responsible for providing a clean, minimal syntax for defining frequent tasks that we run on our remote servers. Using Blade style syntax, one can quickly arrange tasks for deployment, Artisan commands, and more. Envoy only provides support for **Mac and Linux**.

**Passport**

Laravel is used to create API authentication to act as a breeze with the help of Laravel passport. It further provides a full **Oauth2** server implementation for Laravel application in a matter of minutes. Passport is usually assembled on top of **League OAuth2** server which is maintained by **Alex Bilbie**.

**Scout**

Laravel Scout is used for providing a simple, driver-based solution for adding full-text search to the eloquent models. Using model observers, Scout automatically keeps search indexes in sync with eloquent records.

**Socialite**

Laravel Socialite is used for providing an expressive, fluent interface to OAuth authentication with Facebook, Twitter, Google, and Linkedln, etc. It controls almost all the boilerplate social authentication code that you are dreading writing.

18) What do you understand by Unit testing?

Unit testing is built-in testing provided as an integral part of Laravel. It consists of unit tests which detect and prevent regressions in the framework. Unit tests can be run through the available **artisan** command-line utility.

19) What do you know about Facades in Laravel? Explain.

Laravel Facades provide static-like interface classes which are available in the application's service container. Laravel self-ships with several available facades, gives access to almost all features of Laravel. Facades also help to access a service directly from the container itself. It is described in the Illuminate\Support\Facades namespace. Hence, it is easy to use.

**Example**

1. use Illuminate\Support\Facades\Cache;
2. Route::get('/cache', function () {
3. **return** Cache::get('PutkeyNameHere');
4. })

20) How can we check the Laravel current version?

One can easily check the current version of Laravel installation using the **-version** option of artisan command.

1. Php artisan -version

21) How will you explain dd() function in Laravel?

dd stands for "**Dump and Die**." Laravel's dd() function can be defined as a helper function, which is used to dump a variable's contents to the browser and prevent the further script execution.

**Example**

1. dd($array);

22) What do you know about PHP artisan? Mention some artisan command.

PHP artisan is a command-line interface/tool provided with Laravel. It consists of several useful commands which can be helpful while building an application. There are few artisan commands given below:

**PHP artisan list**

A 'list' command is used to view a list of all available Artisan commands.

**PHP artisan help**

Every command also contains a 'help' screen, which is used to display and describe the command's available arguments and options. To display a help screen, run 'help' command.

**PHP artisan tinker**

Laravel's artisan tinker is a repl (**Read-Eval-Print Loop**). Using tinker, one can write actual PHP code through command-line. One can even update or delete table records in the database.

**PHP artisan -version**

By using this command, one can view the current version of Laravel installation.

**PHP artisan make model model\_name**

This command creates a model 'model\_name.php' under the 'app' directory.

**PHP artisan make controller controller\_name**

This command is used to build a new controller file in app/Http/Controllers folder.

23) How will you explain Events in Laravel?

An event is an activity or occurrence recognized and handled by the program. Events in Laravel provide simple observer implementations which allow us to subscribe and listen for events within our application. The event classes are stored in app/Events, while their listeners are stored in app/Listeners of our application. These can be generated using Artisan console commands. A single event may contain multiple listeners that do not depend on each other.

There are some events examples in Laravel which are:

* A new user is registered.
* A new comment is posted.
* User login/logout.
* A new product is added.

24) What are the validations in Laravel?

Validations are approaches that Laravel use to validate the incoming data within the application.

They are the handy way to ensure that data is in a clean and expected format before it gets entered into the database. Laravel consists of several different ways to validate the incoming data of the application. By default, the base controller class of Laravel uses a **ValidatesRequests** trait to validate all the incoming HTTP requests with the help of powerful validation rules.

25) What do you understand by Lumen?

Lumen is a PHP micro-framework built on Laravel's top components. It is created by **Taylor Otwell** (creator of Laravel). It is created for building Laravel based micro-services and blazing fast APIs. It is one of the fastest micro-frameworks available. Lumen is not a complete web framework like Laravel and used for creating APIs only. Therefore, most of the components, such as HTTP sessions, cookies, and templating, are excluded from Lumen. **Lumen** provides support for features such as logging, routing, caching queues, validation, error handling, middleware, dependency injection, controllers, blade templating, command scheduler, database abstraction, the service container, and Eloquent ORM, etc.

One can install Lumen using composer by running the command given below:

1. composer create-project --prefer-distlaravel/lumen blog

26) Which template engine is used by Laravel?

The **blade** is a simple but powerful templating engine provided with Laravel. There is no restriction to use PHP codes in the views. All the blade views are compiled into simple PHP code and cached until they are modified. Blade adds effectively zero overhead to our application. Blade view files in Laravel use **the.blade.php** file extension and are saved in the **resources/views** directory.

27) Explain the Service container and its advantages.

Service container in Laravel is one of the most powerful features. It is an important, powerful tool for resolving class dependencies and performing dependency injection in Laravel. It is also known as **IoC container**.

Dependency injection is a term which essentially means that class dependencies are "injected" into the class by the constructor or, in some cases," setter" methods.

**Advantages of Service Container**

* It can handle class dependencies on object creation.
* It can combine interfaces to concrete classes.

28) What do you know about Laravel Contracts?

Laravel's Contracts are the set of interfaces which are responsible for defining the core functionality of services provided by the Laravel framework.

29) How will you explain homestead in Laravel?

Homestead is an official, pre-packaged, vagrant virtual machine which provides Laravel developers all the necessary tools to develop Laravel out of the box. It also includes **Ubuntu, Gulp, Bower**, and other development tools which are useful in developing full-scale web applications. It provides a development environment which can be used without the additional need to install PHP, a web server, or any other server software on the machine.

30) What are the differences between Laravel and Codeigniter?

|  |  |
| --- | --- |
| **Laravel** | **Codeigniter** |
| Laravel is a framework with an expressive, elegant syntax. | Codeigniter is a powerful framework based on PHP. |
| Laravel is built for the latest version of PHP. | Codeigniter is an older, more mature framework. |
| Laravel is more object-oriented as compared to Codeigniter. | Codeigniter is less object-oriented as compared to Laravel. |
| Laravel can produce model-view-controller, active-record, observer, dependency injection, singleton, restful, façade, event-driven, MTV, and HMVC design patterns. | Codeigniter can produce active-record, model-view-controller, and HMVC design patterns. |
| Laravel supports ORM(object relational mapping). | Codeigniter does not support ORM(object relational mapping). |
| Laravel needs 1 GB memory. | Codeigniter needs 256 GB memory. |
| Laravel has built-in user authentication support. | Codeigniter does not have in-built user authentication support. |

31) How can we get the user's IP address in Laravel?

We can get the user's IP address using:

1. **public** function getUserIp(Request $request){
2. // Gettingip address of remote user
3. **return** $user\_ip\_address=$request->ip();
4. }

32) How can we use the custom table in Laravel?

We can easily use custom table in Laravel by overriding protected $table property of Eloquent. Here, is the sample:

1. **class** User **extends** Eloquent{
2. **protected** $table="my\_user\_table";
3. }

33) What is the use of the Eloquent cursor() method in Laravel?

The cursor method allows us to iterate through our database using a cursor, which will only execute a single query. While processing large amounts of data, the cursor method may be used to reduce your memory usage greatly.

**Example**

1. foreach (Product::where('name', 'bar')->cursor() as $flight) {
2. //make some stuff
3. }

34) How will you create a helper file in Laravel?

We can create a helper file using composer as per the given below steps:

Make a file "app/helpers.php" within the app folder.

Add

1. "files": [
2. "app/helpers.php"
3. ]

in the "autoload" variable.

Now update composer.json with composer dump-autoload or composer update.

35) What are the requirements for Laravel 5.8?

* PHP Version>=7.1.3
* OpenSSL PHP Extension
* PDO PHP Extension
* Mbstring PHP Extension
* Tokenizer PHP Extension
* XML PHP Extension
* Ctype PHP Extension
* JSON PHP Extension

36) In which directory controllers are kept in Laravel?

Controllers are kept in **app/http/Controllers** directory.

37) What is the use of PHP compact function?

PHP compact function receives each key and tries to search a variable with that same name. If a variable is found, then it builds an associate array.

38) What are the major differences between Laravel 4 and Laravel 5.x?

The major differences between Laravel 4 and Laravel 5.x are given below:

* The old app/models directory is entirely removed.
* Controllers, middleware, and requests (a new class in Laravel 5.0) are now combined under the app/Http directory.
* A new app/Providers directory changes the app/start files from previous versions of Laravel of 4.x.
* Application language files and views are moved to the resources directory.
* All major Laravel components include interfaces that are located in the illuminate/contracts repository.
* Laravel 5.x now supports HTTP middleware. The included authentication and CSRF "filters" are converted to middleware.
* One can now type-hint dependencies on controller methods.
* User authentication, registration, and password reset controllers are now combined out of the box, including simple related views which are located at resources/views/auth.
* One can now define events as objects instead of simply using strings.
* Laravel 5 also allows us to represent our queued jobs as simple command objects in addition to the queue job format, which was supported in Laravel 4. These commands are available inside the app/Commands display.

39) Explain some benefits of Laravel over other PHP frameworks.

There are few benefits of Laravel which can be considered over other PHP frameworks:

* In Laravel, Setup and customization process is fast and easy as compared to others.
* Laravel supports multiple file systems.
* It has pre-loaded packages like **Laravel Socialite, Laravel cashier, Laravel Passport, Laravel elixir, and Laravel Scout**, etc.
* It consists of in-built Authentication System.
* It supports Eloquent ORM (Object Relation Mapping) with PHP active record implementation.
* "Artisan" command-line tool for creating a database structure, code skeleton, and build their migration.

40) Which types of relationships are available in Laravel Eloquent?

Below are the types of relationships that Laravel Eloquent ORM supports:

* One to One
* One to Many
* One to Many (Inverse)
* Many to Many
* Has Many Through
* Polymorphic Relations
* Many To Many Polymorphic Relations

41) What do you understand by ORM?

ORM stands for **Object-Relational Mapping**. It is a programming technique which is used to convert data between incompatible type systems in object-oriented programming languages.

42) How can we implement a package in Laravel?

We can implement a package in Laravel by:

* Creating a package folder and name it.
* Creating **Composer.json** file for the package.
* Loading package through main composer.json and PSR-4.
* Creating a Service Provider.
* Creating a Controller for the package.
* Creating a Routes.php file.

43) What do you know about Traits in Laravel?

PHP Traits is a group of methods which can be included within another class. A Trait cannot be instantiated by itself like an abstract class. Traits are generated to reduce the limitations of single inheritance in PHP. It allows a developer to reuse sets of methods freely in various independent classes living in different class hierarchies.

**Example**

1. trait Sharable {
2. **public** function share($item)
3. {
4. **return** 'share this item';
5. }
6. }

We can then include this Trait within other classes like:

1. **class** Post {
2. use Sharable;
3. }
4. **class** Comment {
5. use Sharable;
6. }

Now, if we want to create new objects out of these classes, we would find that they both have the share() method available:

1. $post = **new** Post;
2. echo $post->share(''); // 'share this item'
3. $comment = **new** Comment;
4. echo $comment->share(''); // 'share this item'

44) How can someone change the default database type in Laravel?

Laravel is configured to use MySQL by default.

To change its default database type, edit the file **config/database.php**:

* Search for 'default' =>env('DB\_CONNECTION', 'mysql')
* Change it to whatever required like 'default' =>env('DB\_CONNECTION', 'sqlite')

By using it, MySQL changes to SQLite.

45) How can we use maintenance mode in Laravel 5?

When an application is in maintenance mode, a custom view is displayed for all requests into the application. It makes it easy to "disable" application while it is updating or performing maintenance. A maintenance mode check is added in the default middleware stack for our application. When an application is in maintenance mode, a **MaintenanceModeException** will be thrown with a status code of 503.

We can enable or disable maintenance mode in Laravel 5, simply by executing the below command:

1. // Enable maintenance mode
2. php artisan down
4. // Disable maintenance mode
5. php artisan up

46) How can we create a record in Laravel using eloquent?

We need to create a new model instance if we want to create a new record in the database using Laravel eloquent. Then we are required to set attributes on the model and call the save() method.

**Example**

1. **public** functionsaveProduct(Request $request )
2. $product = **new** product;
3. $product->name = $request->name;
4. $product->description = $request->name;
5. $product->save();

47) How can we check the logged-in user info in Laravel?

**User()** function is used to get the logged-in user

**Example**

1. **if**(Auth::check()){
2. $loggedIn\_user=Auth::User();
3. dd($loggedIn\_user);
4. }

48) How will you describe Fillable Attribute in a Laravel model?

In eloquent ORM, $fillable attribute is an array containing all those fields of table which can be filled using mass-assignment.

Mass assignment refers to sending an array to the model to directly create a new record in Database.

**Code Source**

1. **class** User **extends** Model {
2. **protected** $fillable = ['name', 'email', 'mobile'];
3. // All fields inside $fillable array can be mass-assigned
4. }

49) How will you explain Guarded Attribute in a Laravel model?

The guarded attribute is the opposite of fillable attributes.

In Laravel, fillable attributes are used to specify those fields which are to be mass assigned. Guarded attributes are used to specify those fields which are not mass assignable.

**Code Source**

1. **class** User **extends** Model {
2. **protected** $guarded = ['role'];
3. // All fields inside the $guarded array are not mass-assignable
4. }

If we want to block all the fields from being mass-assigned, we can use:

1. **protected** $guarded = ['\*'];

$fillable serves as a "white list" whereas $guarded functions serves like a "black list". One should use either $fillable or $guarded.

50) What do you know about Closures in Laravel?

In Laravel, a Closure is an anonymous method which can be used as a **callback** function. It can also be used as a parameter in a function. It is possible to pass parameters into a Closure. It can be done by changing the Closure function call in the **handle()** method to provide parameters to it. A Closure can access the variables outside the scope of the variable.

**Example**

1. function handle(Closure $closure) {
2. $closure();
3. }
4. handle(function(){
5. echo 'Interview Question';
6. });

It is started by adding a Closure parameter to the handle() method. We can call the handle() method and pass a service as a parameter.

By using **$closure()**; in the handle() method, we tell Laravel to execute the given Closure which will then display the 'Interview Question.'