PHP Interview Questions

1. 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.

2. 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.

3. 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:

4. 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.

5. 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:

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

6. 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.

7. **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.

8. **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.

9. **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.

10. **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 ?>
```

11. 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.

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

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

13. 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.

14. 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 ---] ?>
```

15. 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.

16. 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 '==='.

17. 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.**

18. 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.

19. Name some of the popular frameworks in PHP.

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

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

20.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

21.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.

22.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.

23.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.

24.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.

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

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

- **Indexed Array** – An array with a numeric index is known as the indexed array. Values are stored and accessed in linear fashion.
- **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.
- **Multidimensional Array** – An array containing one or more arrays is known as multidimensional array. The values are accessed using multiple indices.

26.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 –

```
<?php $variable = "name";
$statement = 'My $variable will not print!n';
print($statement);
print ";"
$statement = "My $variable will print!n"
print($statement);?>
```

It will give the following **output**–

My $variable will not print!

My name will print

27.How to concatenate two strings in PHP?

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

```
<?php $string1="Hello edureka"; $string2="123"; echo $string1 . " " .
$string2; ?>
```

This will produce following **result** −

Hello edureka 123

28.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.

29.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.

30.Name some of the functions in PHP.

- **split()** – The split() function will divide a string into various elements, the boundaries of each element based on the occurrence of pattern in string.

31.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.

32.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.

Following example shows how to **create a cookie** in PHP-

```
<?php $cookie_value = "edureka"; setcookie("edureka", $cookie_value, time()+3600, "/your_
(isset($_COOKIE['cookie'])) echo $_COOKIE["edureka"]; ?>
```

Following example shows how to **start a session** in PHP-

```
<?php session_start(); if( isset( $_SESSION['counter'] ) ) { $_SESSION['counter'] += 1; }els
                                                this page". $_SESSION['counter']; $msg .= "in th
```

33.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.

34.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.

35.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.

36.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**

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**.

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

GET	POST

The GET method is restricted to send up to 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.

38.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.

39.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.

40.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()

41.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:

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

42.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.

43.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**–

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

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.

44.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.

45. How can we get the IP address of the client?

`$_SERVER["REMOTE_ADDR"];`

46. What's the difference between `unset()` and `unlink()`

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

47. What is the output of the following code:

```
$a = '1';

$b = &$a;

$b = "2$b";

echo $a.", ".$b;
```

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

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 as they help developers understand what is going on during the development, and what to look out for during debugging.

49.How can you enable error reporting in PHP?

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 one can instantly get the exact line that is producing the error and can see also if the script in general is behaving correctly.

50.What are Traits?

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.

51. Can the value of a constant change during the script's execution?

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

52. Can you extend a Final defined class?

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

53. What are the __construct() and __destruct() methods in a PHP class?

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.

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

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

55. 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.

```
<?php

function showMessage($hello=false){

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

```
}?>
```

56.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`?

```
<?php

echo array_sum(explode(',',$input));

?>
```

The `explode` function is one of the most used functions in PHP

57.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.

```
<?php

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

```
  echo "The email cannot be empty";

}

?>
```

knowledge about forms management and validation.

58.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?

```
<?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;
   }
 }
}
?>
```

knowledge about OOP.

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

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

60. What are getters and setters and why are they important?

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 one is 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 a setter is not being used and the developer is just declaring the `$userName` variable by hand, you could end up with results as such: `"kevin"`, `"KEVIN"`, `"KeViN"`, `""`, etc. With a setter, the developer can not only adjust the value, for example, `ucfirst($userName)`, but 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, it can be modifyed the results to include `strtoupper($userName)` for proper formatting further up the chain.

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

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.

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

    The candidate should not output anything to the browser before using code that modifies the HTTP headers. Once the developer calls `echo` or any other code that clears the buffer, the developer can no longer set cookies or headers. That is also true for error messages, so if an error happens before using the header command and the INI directive `display_errors` is set, then that will also cause that error to show.

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

    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.

What does the following code output?

```
$i = 016;

echo $i / 2;
```

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.

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

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, one should avoid using == as this would also take into account 0/1 or other similar representation.

65. What are PSRs? Choose 1 and briefly describe it.

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.

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

One 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.

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

Using Composer is a tool for dependency management. The candidate can 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 depended on so less time is spent

68. What are the Features of PHP7.4?

Answer: Some new features in PHP7.4 are:

- Preloading to further improve performance.
- Arrays spread operator.
- One line arrow functions (short closures) for cleaner code.
- Typed properties in the class.
- Formatting numeric values using underscores.
- Extension development using FFI.
- A better type of variance.

69.Explain the difference between $message and $$message?

Answer: $message is a regular variable, which has a fixed name and fixed value, whereas a $$message is a reference variable, which stores data about the variable. The value of $$message can change dynamically as the value of the variable changes.

70. Explain magic constants in PHP?

Answer: Magic constants start and end with double underscores and are predefined constants that change their value based on context and usage. There are 9 magic constants in PHP:

__LINE__, __FILE__, __DIR__, __FUNCTION__, __CLASS__, __TRAIT__, __METHOD__, __NAMESPACE__, ClassName::class

71.Explain various data types of PHP?

Answer: There are different data types in PHP:

- **String:** a sequence of characters, e.g.," Hackr.io."
- **Float:** a floating-point (decimal) number, e.g., 23.456
- **Integer:** an integer, e.g. 12
- **Boolean:** represents two states – true, false.
- **Object:** stores values of different data types into single entity, eg. apple = new Fruit();
- **Array:** stores multiple values of same type, eg. array("red", "yellow", "blue")
- **NULL:** when no value is assigned to a variable, it can be assigned NULL. For example, $msg = NULL;

72.Explain isset() function?

Answer: The isset() function checks if the particular variable is set and has a value other than NULL. The function returns Boolean – false if the variable is not set or true if the variable is set. The function can check multiple values: isset(var1, var2, var3…)

73. Explain the various PHP array functions?

Answer: There are many array functions, all of which are part of the PHP core:

Array Functions	Description
array()	creates an array.
array_diff()	compares arrays and returns the differences in the values.
array_keys()	returns all the keys of the array.
array_reverse()	reverses an array.

array_search()	searches a value and returns the corresponding key.
array_slice()	returns the specific parts of the array.
array_sum()	sums all the values of an array.
count()	the number of elements of an array.

Check more functions on the [PHP manual page](#).

74.Explain the difference between indexed and associative array?

Indexed Array	Associative Array

Has numeric keys or indexes.
 	
Indexes start with 0 and are automatically assigned.
 	
example,
 	
`$fruits = array("orange", "apple", banana);`
 	
here, orange is $fruits[0], apple is $fruits[1] and banana is $fruits[2]
	Each key has its value.
	
	Keys are assigned manually and can be strings too.
	
	example,
	
	`$empdetails = array("Sam"=>1200, "Mike" =>1201, "Mac"=>1202);`
	
	here, the individual values can be accessed as,
	
	`$empdetails["Sam"] = "1200";`
	
	Similarly, others

75. Explain various PHP string functions?

Answer: PHP allows for many string operations. Some popular string functions are:

Function	Description	Example Usage

echo()	output one or more string	`echo "Welcome to hackr.io"`

explode()	break string into array	`$mystr = “welcome to hackr.io”`
		
		`explode(“ ”, $mystr)`
		
		will render [0] = “welcome” and so on

ltrim()	removes extra characters or spaces from the left side of the string	`ltrim($mystr, "…. hello")`
parse_str()	Parses a query string into variables	`parse_str("empId=1234&name=Sam");`
str_replace()	replaces specified characters of a string	`str_replace("mysite","hackr.io","Welcome to mysite");`
str_split()	split the string into character array	`str_split("welcome")`

str_word_count()	word count of the string	`str_word_count("my name is sam ");`
		
		`result = 4`

strlen()	calculates length of the string	`strlen("welcome");` result = 7
strncmp()	compare first few characters of a string	`strncmp("welcome to mysite","welcome to hackr.io", 11);` result = 0 if the first 11 characters are same

For more string functions, refer to PHP manual string functions.

76. Explain the difference between require and include?

Answer: Both require and include are constructs and can be called without parentheses: include myfile.php

However, if the file that is to be included is not found, include will issue a warning, and the script will continue to run. Require will give a fatal error, and the script will stop then and there. If a file is critical to the script, then require should be used, else it can be used.

77.Question: How to upload files in PHP?

Answer: Firstly, PHP should allow file uploads; this can be done by making the directive file_uploads = On

You can then add the action method as 'post' with the encoding type as 'multipart/form-data.'

```
<formaction="myupload.php"method="post"enctype="multipart/form-data">
```

The myupload.php file contains code specific to the file type to be uploaded, for example, image, document, etc., and details like target path, size, and other parameters.

You can then write the HTML code to upload the file you want by specifying the input type as 'file.'

78. How to create a database connection and query in PHP?

Answer: To create a database connection:

```
$connection = new mysqli($servername, $username, $password);
where $servername, $username, $password should be defined beforehand by the developer.
To check if the connection was successful:
if ($conn->connect_error) {
 die("Connection error: " . $conn->connect_error);
}
Create database query:
$sql = "CREATE DATABASE PRODUCT";
if ($conn->query($sql) === TRUE) {
 echo "Database successfully created";
} else {
 echo "Error while creating database: " . $conn->error;
}
```

79.Question: Explain Cookies? How to create cookies in PHP?

Answer: Cookies store data about a user on the browser. It is used to identify a user and is embedded on the user's computer when they request a particular page. We can create cookies in PHP using the setcookie() function:

```
setcookie(name, value, expire, path, domain, secure, httponly);
```

here name is mandatory, and all other parameters are optional.

Example,

```
setcookie(“instrument_selected”, “guitar”)
```

80.Explain the Importance of Parser in PHP?

Answer: A PHP parser is a software that converts source code that the computer can understand. This means whatever set of instructions we give in the form of code is converted into a machine-readable format by the parser. You can parse PHP code with PHP using the token_get_all function.

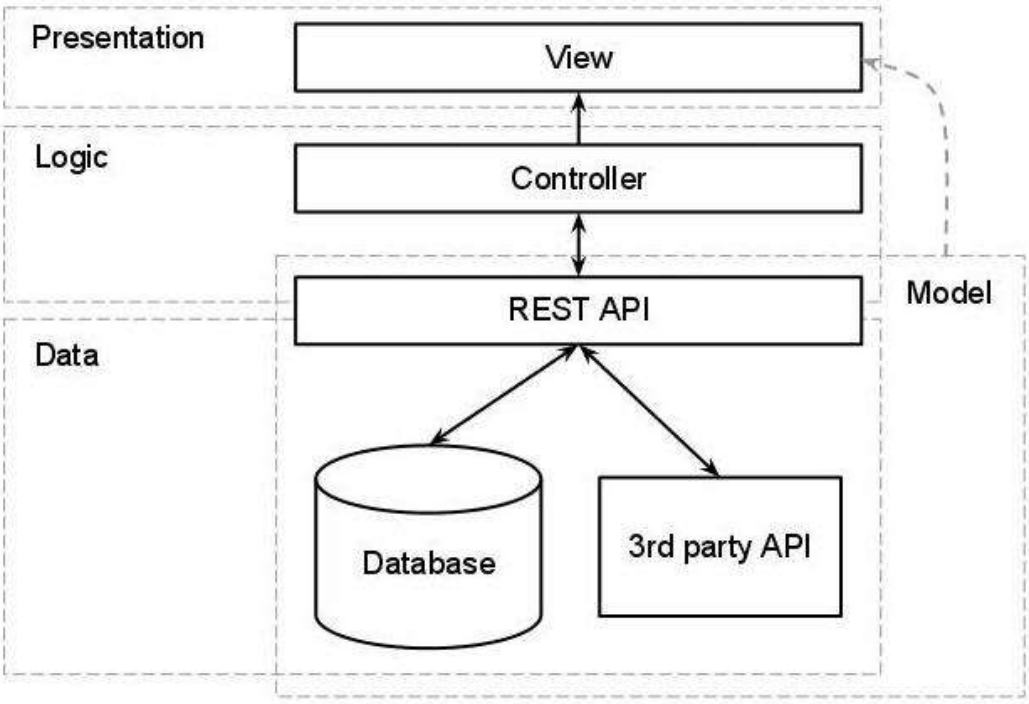
81.Explain the constant() function and its purposes?

Answer: constant() is used to retrieve the value of a constant. It accepts the name of the constant as the input:

```
constant(string $name)
```

The function returns a constant value, if available, else returns null.

82.Can you provide an example of a PHP Web Application Architecture?



r:

83.Define the use of .htaccess and php.ini files in PHP?

Answer: Both of them are used for making changes to the PHP settings.

- .htaccess – A special file that can be used to change or manage the behavior of a website. Directing all users to one page and redirecting the domain's page to https or www are two of the most important uses of the file. For .htaccess to work, PHP needs to be installed as an Apache module.
- php.ini – This special file allows making changes to the default PHP settings. Either the default php.ini file can be edited, or a new file can be created with relevant additions and then saved as the php.ini file. For php.ini to work, PHP needs to run as CGI.

84.Draw a comparison between the compile-time exception and the runtime exception. What are they called?

Answer: Checked exception is the exception that occurs at the compile time. As it is not possible to ignore this type of exception, it needs to be handled cautiously. An unchecked exception, on the other side, is the one that occurs during the runtime. If a checked exception is not handled, it becomes an unchecked exception.

85.Question: Explain Path Traversal?

Answer: Path Traversal is a form of attack to read into the files of a web application. '../' is known as dot-dot-sequences. It is a cross-platform symbol to go up in the directory. To operate the web application file, Path Traversal makes use of the dot-dot-slash sequences.

The attacker can disclose the content of the file attacked using the Path Traversal outside the root directory of a web server or application. It is usually done to gain access token, secret passwords, and other sensitive information stored in the files.

Path Traversal is also known as Directory Traversal. It enables the attacker to exploit vulnerabilities present in the web file under attack.

86.Question: Explain the difference between GET and POST requests.

Answer: Any PHP developer needs to have an adequate understanding of the HTTP protocol. The differences between GET and POST are an indispensable part of the HTTP protocol learning. Here are the major differences between the two requests:

- GET allows displaying the submitted data as part of the URL. This is not the case when using POST as during this time, the data is encoded in the request.
- The maximum number of characters handled by GET is limited to 2048. No such restrictions are imposed on POST.
- GET provides support for only ASCII data. POST, on the other hand, allows ASCII, binary data, as well as other forms of data.
- Typically, GET is used for retrieving data while POST is used for inserting and updating data.

87.Question: Explain the mail function and its syntax.

Answer: To directly send emails from a script or website, the mail() function is used in PHP. It has a total of 5 arguments. The general syntax of a mail function is:

```
mail (to, subject, message, headers, parameters);
```

- to denotes the receiver of the email
- subject denotes the subject of the email
- the message is the actual message that is to be sent in the mail (Each line is separated using /n, and the maximum character limit is 70.)
- headers denote additional information about the mail, such as CC and BCC (Optional)
- parameters denote to some additional parameter to be included in the send mail program (Optional)

88. Question: What is Memcache and Memcached in PHP? Is it possible to share a single instance of a Memcache between several PHP projects?

Answer: Memcached is an effective caching daemon designed specifically for decreasing database load in dynamic web applications. Memcache module offers a handy procedural and object-oriented interface to Memcached.

Memcache is a memory storage space, and it is possible to run Memcache on a single or several servers. Hence, it is possible to share a single instance of Memcache between multiple projects.

It is possible to configure a client to speak to a distinct set of instances. Therefore, running two different Memcache processes on the same host is also allowed. Despite running on the same host, both of such Memcache processes stay independent, unless there is a partition of data.

89. Question: How can you update Memcached when changes are made to PHP?

Answer: There are two ways of updating the Memcached when changes are made to the PHP code:

- Proactively Clearing the Cache – This means clearing the cache when an insert or update is made
- Resetting the Cache – Reset the values once the insert or update is made

90. Question: How is the comparison of objects done in PHP?

Answer: The operator '==' is used for checking whether two objects are instanced using the same class and have the same attributes as well as equal values. To test whether two objects are referring to the same instance of the same class, the identity operator '===' is used.

91. Question: How is typecasting achieved in PHP?

Answer: The name of the output type needs to be specified in parentheses before the variable that is to be cast. Some examples are:

- (array) – casts to array
- (bool), (boolean) – casts to Boolean
- (double), (float), (real) – casts to float
- (int), (integer) – casts to integer
- (object) – casts to object
- (string) – casts to string

92.Question: How would you connect to a MySQL database from a PHP script?

Answer: To connect to some MySQL database, the mysqli_connect() function is used. It is used in the following way:

```
<!--?php $database = mysqli_connect("HOST", "USER_NAME", "PASSWORD");
mysqli_select_db($database,"DATABASE_NAME"); ?-->
```

93.Question: What are constructors and destructors in PHP? Can you provide an example?

Answer: The constructors and destructors in PHP are a special type of functions, which are automatically called when a PHP class object is created and destroyed, respectively.

While a constructor is used to initialize the private variables of a class, a destructor frees the resourced created or used by the class.

Here is a code example demonstrating the concept of constructors and destructors:

```
<?php

class ConDeConExample {

 private $name;

 private $link;

 public function __construct($name) {

 $this->name = $name;

 } # Constructor

 public function setLink(Foo $link){

 $this->link = $link;

 }

 public function __destruct() {

 echo 'Destroying: ', $this->name, PHP_EOL;

 } # Destructor
```

```
}

?>
```

94.Question: What are some common error types in PHP?

Answer: PHP supports three types of errors:

- Notices – Errors that are non-critical. These occur during the script execution. Accessing an undefined variable is an instance of a Notice.
- Warnings – Errors that have a higher priority than Notices. Like with Notices, the execution of a script containing Warnings remains uninterrupted. An example of a Notice includes a file that doesn't exist.
- Fatal Error – A termination in the script execution results as soon as such an error is encountered. Accessing a property of a non-existing object yields a Fatal Error.

95.Question: What are the most important advantages of using PHP 7 over PHP 5?

Answer:

- Support for 64-bit Integers – While PHP 7 comes with built-in support for native 64-bit integers and also for large files, PHP 5 doesn't provide support for either of them.
- Performance – PHP 7 performs far better than PHP 5. PHP 7 uses PHP-NG (NG stands for Next Generation), whereas PHP 5 relies on Zend II.
- Return Type – One of the most important downsides of PHP 5 is that it doesn't allow for defining the return type of a function. This limitation is eliminated by PHP 7, which allows a developer to define the type of value returned by any function in the code.
- Error Handling – It is extremely difficult to manage fatal errors in PHP 5. PHP 7, on the opposite, comes with a new Exception Objects engine. It helps in managing several major critical errors, which are now replaced with exceptions.
- Anonymous Class – To execute the class only once, for increasing the execution time, we have the anonymous class in PHP 7. It is not available in PHP 5.
- Group Use Declaration – PHP 7 allows all the classes, constants, and functions to be imported from the same namespace to be grouped in a single-use statement. This is known as group use declaration. The feature is not available in PHP 5.
- New Operators – Several new operators are introduced to PHP 7, including '<=>' and '??' The former is known as the three-way comparison operator, and the latter is called the null coalescing operator.

96.Question: What are the various ways of handling the result set of Mysql in PHP?

Answer: There are four ways of handling the result set of Mysql in PHP:

- mysqli_fetch_array

- mysqli_fetch_assoc
- mysqli_fetch_object
- mysqli_fetch_row

97. Question: What are the Traits in PHP?

Answer: The mechanism allows for the creation of reusable code in PHP-like languages where there is no support for multiple inheritances. A trait can't be instantiated on its own.

98. Question: What is a session in PHP? Write a code example to demonstrate the removal of session data.

Answer: The simplest way to store data for individual users against a unique session ID is by using a PHP session. It is used for maintaining states on the server as well as sharing data across several pages. This needs to be done because HTTP is a stateless protocol.

Typically, session IDs are sent to the browser using session cookies. The ID is used for retrieving existing session data. If the session ID is not available on the server, then PHP creates a new session and then generates a new session ID.

Here is the program for demonstrating how session data is removed:

```
<?php

session_start();

$_SESSION['user_info'] = ['user_id' =>1,

'first_name' =>

'Hacker', 'last_name' =>

'.io', 'status' =>

'active'];

if (isset($_SESSION['user_info']))

{

echo "logged In";

}

unset($_SESSION['user_info']['first_name']);
```

```
session_destroy(); // Removal of entire session data

?>
```

99. Question: What would you say is the best hashing method for passwords?

Answer: Rather than using the typical hashing algorithms, including md5, sha1, and sha256, it is preferable to use either crypt() or hash(). While crypt() provides native support for several hashing algorithms, hash(,) provides support for even more of them.

100. Question: Why is it not possible for JavaScript and PHP to interact directly? Do you know any workarounds?

Answer: No direct interaction is possible between JS and PHP because while the former is a client-side language, the latter is a server-side language. An indirect interaction between the two leading programming languages can happen using exchanging variables.

This exchange of variable is possible due to two reasons:

- PHP can generate JavaScript code meant to be executed by the browser
- It is achievable to pass a specific variable back to PHP via the URL. Because PHP always gets to be executed before JavaScript, the JS variables must be passed via a form or the URL. To pass the variables, GET and POST are used. Similarly, to retrieve the passed variable, $_GET and $_POST are used.

101. Question: Write code in PHP to calculate the total number of days between two dates?

Answer:

```
<?Php

$date1 = '2019-01-11'; # Date 1

$date2 = '2019-01-09'; # Date 2

$days = (strtotime($date1)-strtotime($date2))/(60*60*24);

echo $days;

?>
```

Output:
1

102. Question: Briefly explain PHP & some of the popular PHP frameworks.

Answer: PHP is a popular server-side scripting language used by developers to dynamically create webpages. Originally, PHP meant Personal Home Page. However, today it stands for the recursive acronym PHP: Hypertext Preprocessor.

As of now, there is a wide range of PHP frameworks available. Three of the most popular PHP frameworks are briefly explained as follows:

- CodeIgniter – Simplistic and powerful, CodeIgniter is an incredibly lightweight PHP framework with a hassle-free installation process and minimalistic configuration requirements. The complete framework is a mere 2 MB and that too, including the documentation. As it comes with many prebuilt modules that help in developing strong, reusable components, CodeIgniter is ideal for developing dynamic websites. The popular PHP framework also offers a smooth working experience on both dedicated as well as shared hosting platforms.
- Laravel – Although not as old as some of the other popular PHP frameworks, Laravel is perhaps the most popular PHP framework. Launched in 2011, the immense popularity enjoyed by the PHP framework can be credited to its ability to offer additional speed and security for handling complex web applications. Laravel also eases the development process by reducing the complexity of repetitive tasks, including authentication, routing, sessions, and queuing.
- Symfony – Used by PHP developers ever since its release back in 2005, Symfony is a popular PHP framework that has stood the test of time. It has matured over its run of almost one-and-a-half decade. An extensive PHP framework, Symfony, is the sole PHP framework that follows the standard of PHP and web completely. Popular CMSs like Drupal, OroCRM, and PHPBB make use of various Symfony components.

If you are interested in learning Codeigniter, Laravel or Symfony, then Hackr has programming community-recommended best tutorials and courses:

- Codeigniter tutorials and courses
- Laravel tutorials and courses
- Symfony tutorials and courses

103. Question: Will you draw a comparison between server-side and client-side programming languages?

Answer: Server-side programming languages are used for building programs that run on a server and generate the contents of a webpage. Examples of server-side programming languages include C++, Java, PHP, Python, and Ruby. A server-side programming language helps in:

- Accessing and/or writing a file present on the server
- Interacting with other servers
- Processing user input
- Querying and processing the database(s)
- Structuring web applications

Contrary to the server-side programming languages, client-side programming languages help in developing programs that run on the client machine, typically browser, and involves displaying output and/or additional processing, such as reading and writing cookies.

CSS, HTML, JavaScript, and VBScript are popular client-side programming languages. A client-side programming language allows:

- Developing interactive webpages
- Interacting with temporary storage and/or local storage
- Making data and/or other requests to the server
- Offering an interface between the server and the end-user

104. Question: Please explain the differences between the echo and print statements in PHP?

Answer: There are two statements for getting output in PHP - echo and print. Following are the distinctions between the two:

- Although used rarely, echo has the ability to take multiple parameters. The print statement, on the contrary, can accept only a single argument
- Echo has no return value whereas, print has a return value of 1. Hence, the latter is the go-to option for using in expressions
- Generally, the echo statement is preferred over the print statement as it is slightly faster

105. Question: How do static websites differ from dynamic websites?

Answer: There are two types of websites, static and dynamic. Differences between the two are enumerated as follows:

- Advantage – The main advantage of a static website is flexibility, whereas the main advantage for a dynamic website comes in the form of a CMS
- Changes/Modification – Changes are made to the content of a static website only when the file is updated and published i.e., sent to the webserver. Dynamic websites, on the other hand, contains “server-side” code that allows the server to generate unique content when the webpage is loaded
- Content – The content remains the same every time the page is reloaded for a static website. Contrarily, the content belonging to a dynamic website is updated regularly
- Response – Static websites send the same response for every request whereas dynamic websites might generate different HTML for different requests

- Technologies Involved – Mere HTML is used for building static websites while dynamic websites are developed using several technologies, such as ASP.Net, JSP, Servlet, and PHP

106. Question: Please explain the use of the imagetypes() function?

Answer: The imagetypes() function gives the image format and types supported by the present version of the GD-PHP.

107. Question: Can you explain the difference between 'passing the variable by value' and 'passing the variable by reference' in PHP?

Answer: Passing the variable by value means that the value of the variable is directly passed to the called function. It then uses the value stored in the variable. Any changes made to the function doesn't affect the source variable.

Passing the variable by reference means that the address of the variable where the value is stored is passed to the called function. It uses the value stored in the passed address. Any changes made to the function does affect the source variable.

108. Question: What do you understand by typecasting and type juggling?

Answer: When a variable's data type is converted explicitly by the user, it is known as typecasting. The PHP programming language doesn't support explicit type definition in variable declaration. Hence, the data type of the variable is determined by the context in which the variable is used.

For example, if a string value is assigned to a $var variable, then it automatically gets converted to a string. Likewise, if an integer value is assigned to $var, then it becomes an integer. This is called type juggling.

109. Question: Could you explain how to fetch data from a MySQL database using PHP?

Answer: To begin with, you need to first establish a connection with the MySQL database that you wish to use. For that, you can use the mysqli_connect() function.

Suppose that the database that you need to access is stored on a server named localhost and has the name instanceDB. Also, it has user_name as the username and pass_word as the password.

For establishing a connection to the instanceDB, you need to use the following PHP code:

```
<?php
$servername = "localhost";
$username = "user_name";
$password = "pass_word";
$dbname = "instanceDB";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
if (!$conn) { // For checking connection to the database
 die("Connection failed: " . mysqli_connect_error());
}
```

Next, you need to use the SELECT statement to fetch data from one or more tables. The general syntax is:

```
SELECT column_name from table_name
```

Suppose, we have a single table called instancetable with column_1, column_2, and column_3 in the instanceDB, then to fetch data; we need to add the following PHP code:

```
$sql = "SELECT column_1, column_2, column_3 from instancetable";
$result = $conn->query($sql);
```

110. Question: How will you display text with PHP Script?

Answer: Either the echo statement or the print statement can be used for displaying text with a PHP script. In usual scenarios, the former is preferred over the latter because it is slightly faster.

111. Question: What are some of the most popular PHP-based Content Management Systems (CMS)?

Answer: There are a plethora of PHP-based Content Management Systems in use today. Drupal, Joomla, and WordPress are the most popular among the bunch.

112. Question: Can you explain PHP parameterized functions?

Answer: Functions with parameters are known as PHP parameterized functions. It is possible to pass as many parameters as you'd like inside a function. Specified inside the parentheses after the function name, these all parameters act as variables inside the PHP parameterized function.

113. Question: Explain $_SESSION in PHP?

Answer: The $_SESSION[] is called an associative array in PHP. It is used for storing session variables that can be accessed during the entire lifetime of a session.

114. Question: Explain the difference between substr() and strstr() functions?

Answer: The substr() function returns a part of some string. It helps in splitting a string part-by-part in PHP. This function is typically available in all programming languages with almost the same syntax.

General syntax:

```
substr(string, start, length);
```

The strstr() function is used for searching a string within another string in PHP. Unlike the substr() function, strstr() is a case-sensitive function.

General syntax:

```
strstr(string, search, before_string);
```

115. Question: Explain the use of the $_REQUEST variable?

Answer: $_REQUEST is an associative array that, by default, contains the contents of the $_COOKIE, $_GET, $_POST superglobal variables.

Because the variables in the $_REQUEST array are provided to the PHP script via COOKIE, GET, and POST input mechanisms, it could be modified by the remote user. The variables and their order listed in the $_REQUEST array is defined in the PHP variables_order configuration directive.

116. Question: Please enumerate the major differences between for and foreach loop in PHP?

Answer: Following are the notable differences between for and for each loop:

- The for-each loop is typically used for dynamic array
- The for loop has a counter and hence requires extra memory. The for-each loop has no counter, and hence there is no requirement for additional memory
- You need to determine the number of times the loop is to be executed when using the for a loop. However, you need not do so while using the for each loop

117. Question: Is it possible to submit a form with a dedicated button?

Answer: Yes, it is possible to submit a form with a dedicated button by using the document.form.submit() method. The code will be something like this:

```
<input type=button value="SUBMIT" onClick="document.form.submit()">
```

118. Question: Please explain whether it is possible to extend a final defined class or not?

Answer: No, it isn't possible to extend a final defined class. The final keyword prevents the class from extending. When used with a method, the final keyword prevents it from overriding.

119. Question: Will it be possible to extend the execution time of a PHP script? How?

Answer: Yes, it is possible to extend the execution time of a PHP script. We have the set_time_limit(int seconds) function for that. You need to specify the duration, in seconds, for which you wish to extend the execution time of a PHP script. The default time is 30 seconds.

References:

https://www.edureka.co/blog/interview-questions/php-interview-questions/

https://arc.dev/interview/php-interview-questions-and-answers

https://hackr.io/blog/php-interview-questions