1. How require() function differs from require_once?

The primary purpose of both functions is to include PHP files into other PHP files. But, both functions have a minor difference. Let's understand that difference:
Require(): This function allows us to include files multiple times in a single PHP file. You have to take care of the file's location because if the function does not find the file on the mentioned locations, it throws error and halts the execution.
Require_once(): The function allows us to include files multiple times in a single PHP file, but the PHP server includes that file only once. In simple terms, after including a file the first time, if you again include it, the compiler ignores it. Just like the Require() function, you have to take care of the file's location to prevent any fatal error and execution halt.

2. What is the difference between session_start() and session_destroy() function?

PHP session() allows you to store and use the information on different pages. You should always remember that as soon as the browser is closed, session() automatically gets destroyed. Let’s see how session_start() is different from the session_destroy():

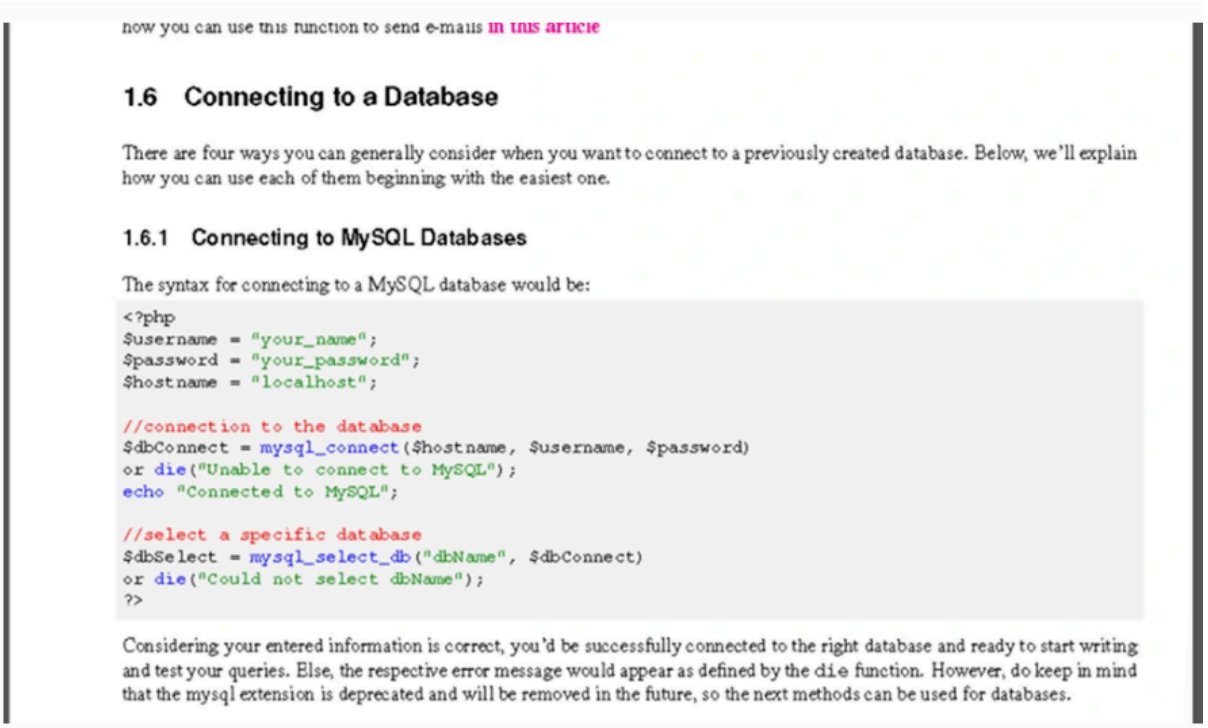
Session_start()	Session_destroy()
It creates or resumes the session.	It destroys the session.
Its main job is creating a session or resuming the current one according to a session identifier passed through a GET, POST request, or a cookie.	Its main job is to destroy all the information associated with the current session. It does not unset any of the global variables associated with the session or unset the session cookie.
It has one parameter, which is optional. Its syntax: Session_start (array $options = []): bool	It does not have any parameters. It's the syntax: Session_destroy()
It returns true if the session starts successfully; otherwise, it returns false.	It returns true on success or false on failure.

3. What are the major steps for creating a new database using MySQL and PHP?

It is easy to create a new MySQL database in PHP. By executing a query, you can create a database. Here is how the process works:

First step: Connection establishment
Establish the connection to the MySQL server using PHP script. The server establishes and validates the connection.
Second step: Write queries
Write a SQL query for further verification after validation in the first step. Now, write other queries for creating the database and storing them into string variables.
Third step: Execution
Executes queries one by one to complete the process of database creation.

Check out the syntax in the below image:



now you can use this function to send e-mails in this article

1.6 Connecting to a Database

There are four ways you can generally consider when you want to connect to a previously created database. Below, we'll explain how you can use each of them beginning with the easiest one.

1.6.1 Connecting to MySQL Databases

The syntax for connecting to a MySQL database would be:

```
<?php
$username = "your_name";
$password = "your_password";
$hostname = "localhost";

//connection to the database
$dbConnect = mysql_connect($hostname, $username, $password)
or die("Unable to connect to MySQL");
echo "Connected to MySQL";

//select a specific database
$dbSelect = mysql_select_db("dbName", $dbConnect)
or die("Could not select dbName");
?>
```

Considering your entered information is correct, you'd be successfully connected to the right database and ready to start writing and test your queries. Else, the respective error message would appear as defined by the die function. However, do keep in mind that the mysql extension is deprecated and will be removed in the future, so the next methods can be used for databases.

Image Credit: PHP_cookbook and here you can also find the various ways to execute queries

4. What is type hinting in PHP?

As the name suggests, type hinting in PHP is like giving hints to a specific function. PHP 5 brought this concept to specify data types for function arguments in function declarations. The expected data types could be objects, arrays, interfaces, etc. Whenever the function has been called, PHP checks if the arguments are of a user-preferred type or not. The runtime will generate an error if the argument is not of the specified type, and PHP will halt the program execution.

For instance, if you previously mentioned an integer datatype as the datatype of the function arguments in the function declaration and attempt to pass float or other data instead of an integer value, you'll get the error "value must only be integer".

The 2 main benefits of type hinting:

Code optimization

Improved error messages

5. How Get method is different from the Post method?

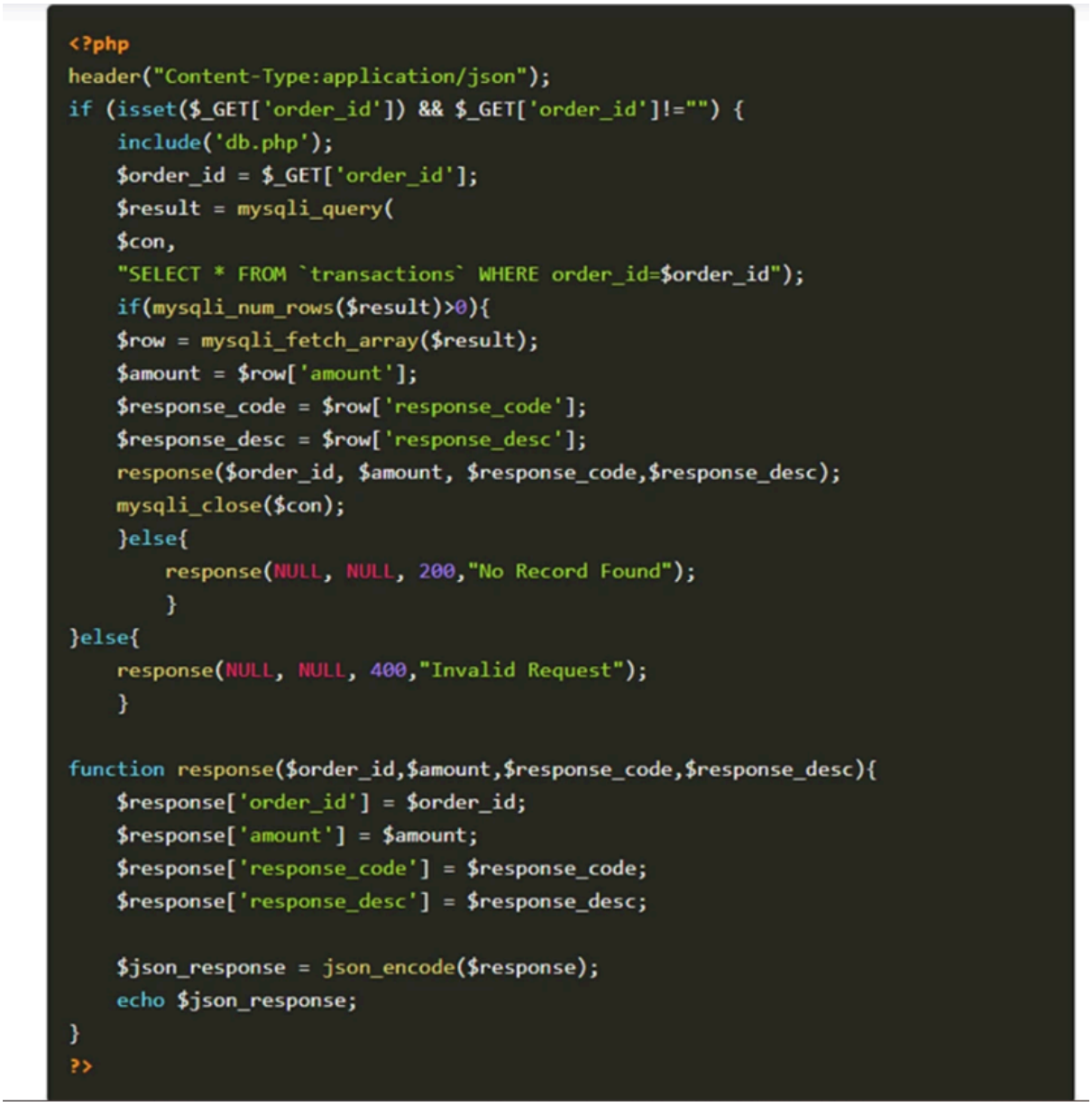
The server-side scripting language, PHP, uses Get and Post methods for sending and receiving the information to and from the server. The following table highlights the difference between Get and Post methods:

Get	Post
It sends the information by sending a request from a specified resource.	It sends information using an HTTP header.
It is the best way to use this to send general information which is non-sensitive.	It is the best way to use it for sending sensitive information.
You can send only limited information (1500 characters) using this method.	You can send unlimited information through this method.
You can easily see information in the URL.	Since information is non-visible in the URL thus, you can't see them.
It is less secure than the Post method.	It is most secure than Get method.
You can bookmark the data and other information.	You can't bookmark any data.

6. How can you create a simple REST API in PHP?

REST stands for Representational State Transfer which refers to system architecture to define methods for accessing web services. The primary function of the REST API is to build a system which can be used by other mobile or web applications. CRUD (Create, Read, Update, Delete) operations have been used for creating Rest API. REST uses Get, Post, Put, and Delete requests to implement these operations. Like Get- Read; Post-Create; Put- Update; Delete-Delete. It generates output data in varied formats such as Extensible Markup Language (XML), JavaScript Object Notation (JSON), and Command Separated Value (CSV).
There are three simple steps for creating Rest API

- Create a database and table (use query – Create a table or Create database)
- Establish a database connection (use query – Connect)
- Create a REST API file. (see the image to know the sample script which has used Get request and output format is JSON)



```php
<?php
header("Content-Type:application/json");
if (isset($_GET['order_id']) && $_GET['order_id']!="") {
    include('db.php');
    $order_id = $_GET['order_id'];
    $result = mysqli_query(
    $con,
    "SELECT * FROM `transactions` WHERE order_id=$order_id");
    if(mysqli_num_rows($result)>0){
    $row = mysqli_fetch_array($result);
    $amount = $row['amount'];
    $response_code = $row['response_code'];
    $response_desc = $row['response_desc'];
    response($order_id, $amount, $response_code,$response_desc);
    mysqli_close($con);
    }else{
        response(NULL, NULL, 200,"No Record Found");
        }
}else{
    response(NULL, NULL, 400,"Invalid Request");
    }

function response($order_id,$amount,$response_code,$response_desc){
    $response['order_id'] = $order_id;
    $response['amount'] = $amount;
    $response['response_code'] = $response_code;
    $response['response_desc'] = $response_desc;

    $json_response = json_encode($response);
    echo $json_response;
}
?>
```

Image Credit: Allphtrick

7. How do you create cookies in PHP?

A cookie is a file that the server embeds on the user's computer to track user information such as username, email address, phone number, etc.

Always remember that, whenever you request a page from the same device, the server sends the cookie as well. In PHP, you can create a cookie using setcookie() function. You must call the cookie before a script generates any outcome.

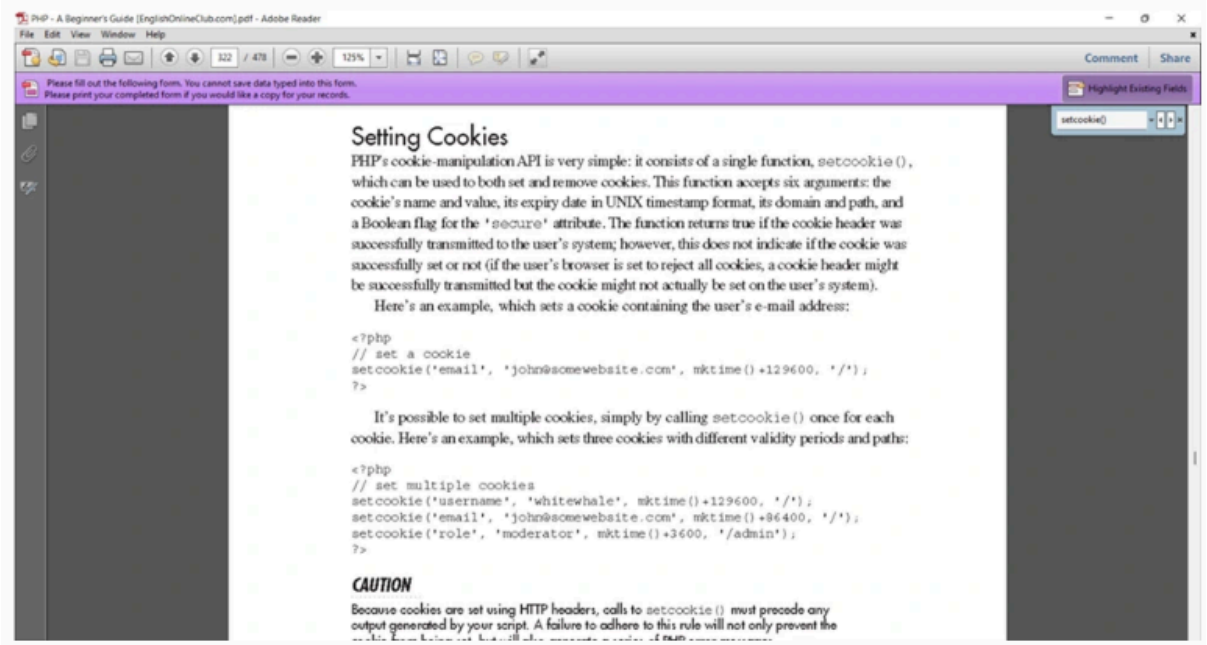
Syntax

setcookie(name, value, expire, path, domain, secure):

Here

- **Name** – mandatory argument to set the name
- **Value** – Optional argument to set the value
- **Expire** – Optional argument to set the expiry timestamp of the cookie
- **Path** – Optional argument to specify cookie path on the server
- **Domain** – Optional argument to specify the domain for which cookie is available
- **Secure** – Optional argument to ensure the secure connection using HTTP only

See the following images to understand various examples of setting cookies. You can see a cookie which contains the user's email address.



Setting Cookies

PHP's cookie-manipulation API is very simple: it consists of a single function, `setcookie()`, which can be used to both set and remove cookies. This function accepts six arguments: the cookie's name and value, its expiry date in UNIX timestamp format, its domain and path, and a Boolean flag for the `'secure'` attribute. The function returns true if the cookie header was successfully transmitted to the user's system; however, this does not indicate if the cookie was successfully set or not (if the user's browser is set to reject all cookies, a cookie header might be successfully transmitted but the cookie might not actually be set on the user's system).

Here's an example, which sets a cookie containing the user's e-mail address:

```
<?php
// set a cookie
setcookie('email', 'john@somewebsite.com', mktime()+129600, '/');
?>
```

It's possible to set multiple cookies, simply by calling `setcookie()` once for each cookie. Here's an example, which sets three cookies with different validity periods and paths:

```
<?php
// set multiple cookies
setcookie('username', 'whitewhale', mktime()+129600, '/');
setcookie('email', 'john@somewebsite.com', mktime()+86400, '/');
setcookie('role', 'moderator', mktime()+3600, '/admin');
?>
```

CAUTION

Because cookies are set using HTTP headers, calls to `setcookie()` must precede any output generated by your script. A failure to adhere to this rule will not only prevent the

Image Credit: PHP Beginner's Guide

8. Discuss runtime exceptions and compile time exceptions.

The exception is an important OOPs concept to define any logical or runtime error. It could be either runtime exception or compile time. In PHP, any class, function, object, or user-defined function can throw exceptions.
Runtime Exception:
This exception is a runtime exception which occurs during runtime. It can happen even after the successful compilation and execution of the codes. The main reasons behind this exception are invalid user inputs, invalid API communications, increased size of storage media, or other similar events.
For example, you have defined an integer variable in code but input a float number at runtime. It would execute codes but would not generate the outcome. You must use the Try-catch block to catch this exception and manage them appropriately.
Compile Time Exception:
This exception is compiled time exception, which occurs during code writing and compiling. It happens because of coding mistakes, or you have made syntax or semantic mistakes in the code. For example, undeclared variables, missing parenthesis or commas, or other similar cases. To execute code successfully, you must catch the compile time exception and fix the code.

9. Can you perform password encryption in PHP?

Yes, you can perform password encryption in PHP using many ways. The primary objective of PHP encryption is to protect the data. It uses hashing algorithms to convert the data into encrypted text. You need to use specific decryption processes to decode the data again.
The function password_hash() is used to protect and verify the passwords.
A generic step of password encryption is as follows:

- Create a Unique Encryption Key (DEK)
- Scramble the information utilizing unique key encryption
- Move the DEK to the cloud for encryption which would return an encrypted data key (KEK)
- Save the KEK and encrypted data

- Use generated key (DEK)

Three ways of password encryption in PHP

- **Hashing** – A simplest way to encrypt passwords using Hash. It takes an input value and converts it into a fixed-length outcome which is called "hash", "hash value", or "message digest". It should be used for storing user passwords.
- **Secret Key Encryption** – Another simple way to encrypt a password using one key only. It is also called symmetric encryption, which uses a shared secret (single key) to encrypt and decrypt the data. This means the sender and receiver use the same key to ensure data privacy.
- **Envelope encryption** – It is not used widely for encryption. The central concept of this encryption is to put the required data in a virtual envelope after encryption, lock it for the receiver(s), put the receiver(s) address, and mention the lock and other key identifiers on the outside of the envelope.

Check this link to get the codes of each encryption.

Databases play a crucial role in PHP. Take a look at this link to find out about different database concepts.

10. How can we terminate the script execution in PHP?

The exit() function is an in-built PHP function which is used to terminate script execution. It outputs a message first and then terminates the script.

- The shutdown functions and object destroyers are independent of the exit() function and they will be executed even if the script is terminated.
- Whatever string is passed to the exit() function, it will be displayed before termination.
- It can be called without parentheses because it is a language construct.

Syntax:

exit(string $status = ?): void

If status is a string, it will be displayed just before termination.

exit(int $status): void

If status is an integer, it will be used as an exit status and will not be printed.

11. How can we connect to a URL in PHP?

To connect to a URL, we can use cURL (Client for URLs), a default library in PHP. The client issues a GET request and receives the information which has been asked, such as headers, HTML, etc. There are cURL functions available to support these actions they are:

- curl_init(): To initialise the cURL session
- curl_setopt(): To set all options for transfer
- curl_exec(): To execute the session
- curl_close(): To close the session

Below example shows use of cURL for downloading contents of a remote website into a local file.

```php
<?php

$handle = curl_init();

$url = "https://www.ladygaga.com";



// Set the url

curl_setopt($handle, CURLOPT_URL, $url);

// Set the result output to be a string.

curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);



$output = curl_exec($handle);



curl_close($handle);



echo $output;

?>
```

12. Why is callback used in PHP?

A callback function is a function which can be passed as an argument into another function. They are denoted by a callable type and their name is passed in the calling function as an argument.
Example: Pass a callback to PHP's array_map function.

```php
<?php

function my_callback($item) {

  return strlen($item);

}

$strings = ["apple", "orange", "banana", "coconut"];

$lengths = array_map("my_callback", $strings);

print_r($lengths);

?> <span data-ccp-props="{'201341983':0,'335559685':720,'335559740':276}">
</span>
```

13. Will it be possible to extend the execution time of PHP?

The maximum execution time is defined by max_execution_time directive.
After PHP reaches the maximum execution time, a fatal error will be generated and PHP will timeout and exit.
Fatal error: Maximum execution time of 30 seconds exceeded in example.php
If the execution time is not set then the default value is 30 seconds.
There are below ways to increase the execution time:

- Configure php.ini
- Use ini_set() function
- Use set_time_limit() function
- Configure .htaccess
- Configure cPanel options

14. Explain major differences between for loop and foreach loop.

For loop	Foreach loop
It is executed over variables	Executed on Arrays - associative and numerical
Executes till the given condition is true	Executes till the given condition is true

Syntax:
for (initialisation; condition; increment)
{
Code to be executed;
}
	Syntax:
	foreach ($array as $value)
	{
	Code to be executed;
	}