### 

### 

### 

### 

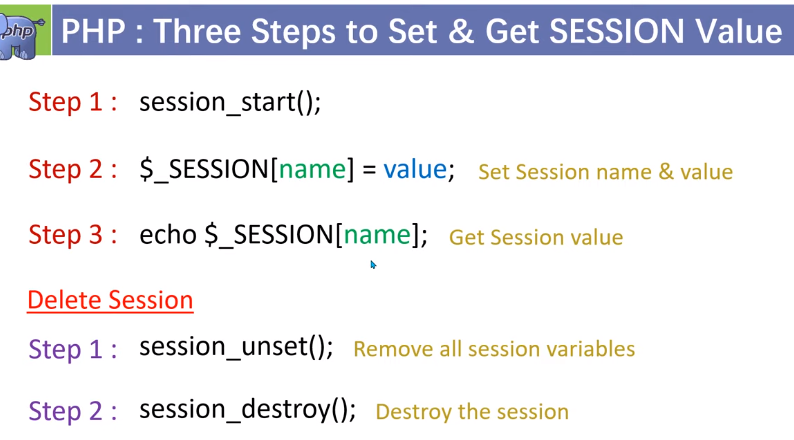
### 

### 

### 

### 

### 

****

### C:\Users\Rajiv\Desktop\Screenshot_1.jpg

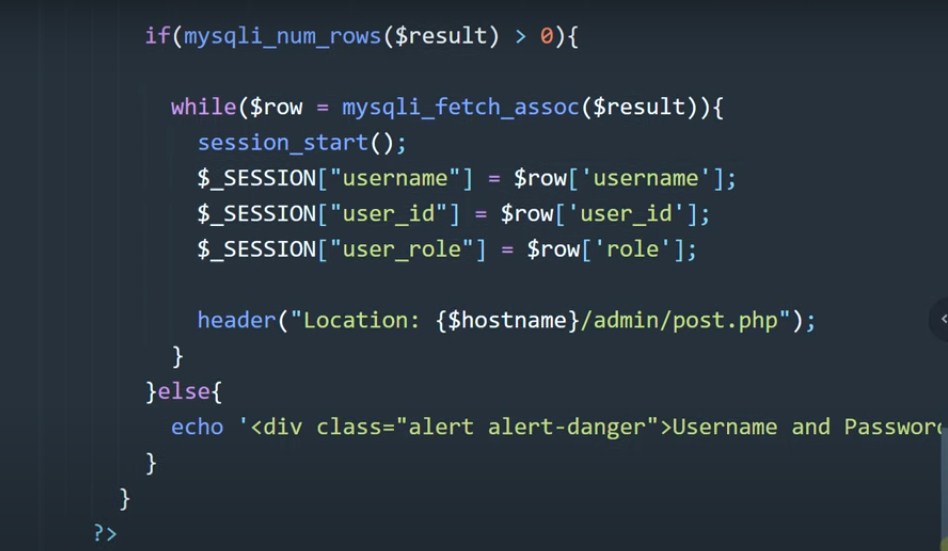
### Caching in PHP SQL

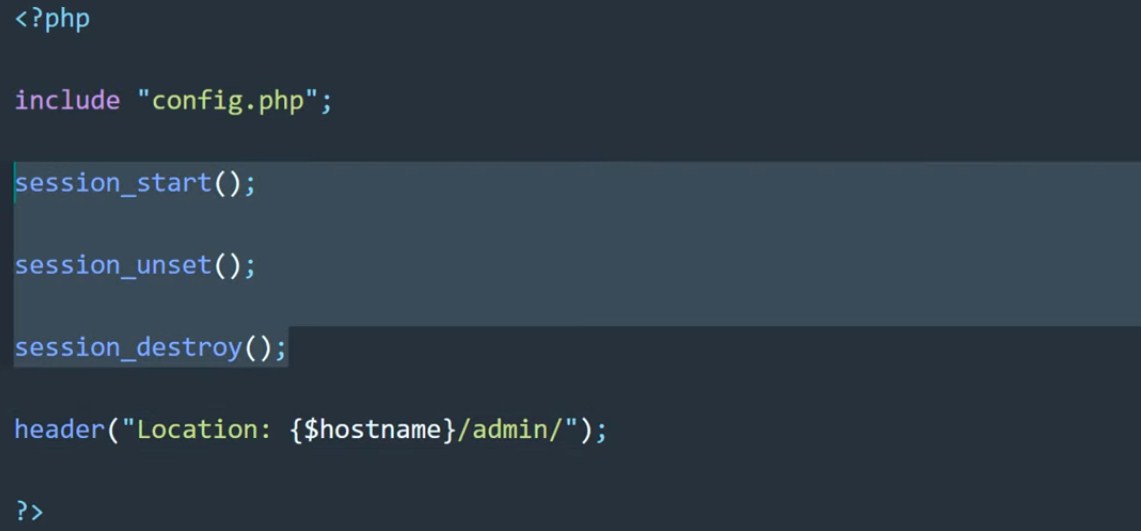
### 

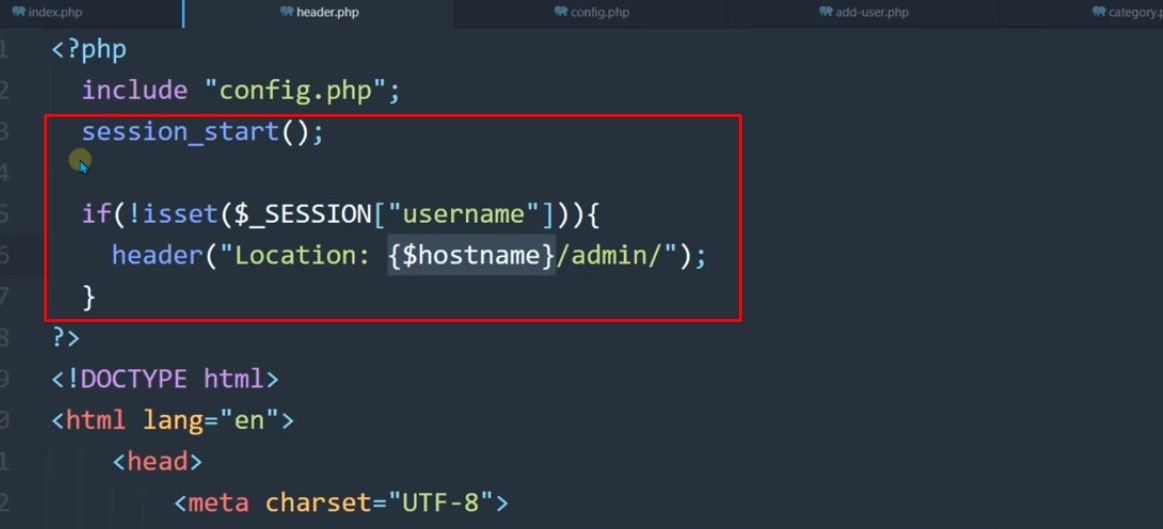
### C:\Users\Rajiv\Desktop\Screenshot_4.jpgC:\Users\Rajiv\Desktop\Screenshot_5.jpg

### C:\Users\Rajiv\Desktop\Screenshot_2.jpgC:\Users\Rajiv\Desktop\Screenshot_3.jpg

### PHP Login logout example with session







### Second method

CREATE TABLE `login\_user` (

`id` int(11) NOT NULL,

`name` varchar(60) NOT NULL,

`user\_name` varchar(20) NOT NULL,

`password` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

#### login.php

<?php

    session\_start();

    $message="";

    if(count($\_POST)>0) {

        $con = mysqli\_connect('127.0.0.1:3306','root','','admin') or die('Unable To connect');

        $result = mysqli\_query($con,"SELECT \* FROM login\_user WHERE user\_name='" . $\_POST["user\_name"] . "' and password = '". $\_POST["password"]."'");

        $row  = mysqli\_fetch\_array($result);

        if(is\_array($row)) {

        $\_SESSION["id"] = $row['id'];

        $\_SESSION["name"] = $row['name'];

        } else {

         $message = "Invalid Username or Password!";

        }

    }

    if(isset($\_SESSION["id"])) {

    header("Location:index.php");

    }

?>

<html>

<head>

<title>User Login</title>

</head>

<body>

<form name="frmUser" method="post" action="" align="center">

<div class="message"><?php if($message!="") { echo $message; } ?></div>

<h3 align="center">Enter Login Details</h3>

 Username:<br>

 <input type="text" name="user\_name">

 <br>

 Password:<br>

<input type="password" name="password">

<br><br>

<input type="submit" name="submit" value="Submit">

<input type="reset">

</form>

</body>

</html>

#### index.php

<?php

session\_start();

?>

<html>

<head>

<title>User Login</title>

</head>

<body>

<?php

if($\_SESSION["name"]) {

?>

Welcome <?php echo $\_SESSION["name"]; ?>. Click here to <a href="logout.php" tite="Logout">Logout.

<?php

}else echo "<h1>Please login first .</h1>";

?>

</body>

</html>

#### logout.php

<?php

session\_start();

unset($\_SESSION["id"]);

unset($\_SESSION["name"]);

header("Location:login.php");

?>

# How to add captcha in PHP form

#### index.php

<?php

session\_start();

    if ($\_POST["vercode"] != $\_SESSION["vercode"] OR $\_SESSION["vercode"]=='')  {

        echo "<script>alert('Incorrect verification code');</script>" ;

    }

    else{

        echo "<script>alert('Verification code match !');</script>" ;

    }

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Elegant Account Login Form with Avatar Icon</title>

<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

<link href="https://fonts.googleapis.com/css?family=Roboto|Varela+Round" rel="stylesheet">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

</head>

<body>

<div class="login-form">

    <form action="" method="post">

        <div class="avatar"><i class="material-icons"></i></div>

        <h4 class="modal-title">Login to Your Account</h4>

        <div class="form-group">

            <input type="text" class="form-control" placeholder="Username" required="required">

        </div>

        <div class="form-group">

            <input type="password" class="form-control" placeholder="Password" required="required">

        </div>

        <div class="form-group">

            <input type="text" name="vercode" class="form-control" placeholder="Verfication Code" required="required">

        </div>

        <div class="form-group small clearfix">

            <label class="checkbox-inline">Verification Code</label>

              <img src="captcha.php" >

        </div>

        <input type="submit" class="btn btn-primary btn-block btn-lg" value="Login">

    </form>

</div>

</body>

</html>

#### captcha.php

<?php

    session\_start();

    $text = rand(10000,99999);

    $\_SESSION["vercode"] = $text;

    $height = 25;

    $width = 65;

    $image\_p = imagecreate($width, $height);

    $black = imagecolorallocate($image\_p, 0, 0, 0);

    $white = imagecolorallocate($image\_p, 255, 255, 255);

    $font\_size = 14;

    imagestring($image\_p, $font\_size, 5, 5, $text, $white);

    imagejpeg($image\_p, null, 80);

?>

### 

### 

### 

### 

### 

### Logout example

### 

# PHP Visitor Count

Index.php

<?php

$servername="localhost";

$username="root";

$password="";

$dbname="traffic";

$conn=mysqli\_connect($servername,$username,$password,"$dbname");

if(!$conn){

die('Could not Connect My Sql:' .mysql\_error());

}

$find\_count=mysqli\_query($conn,"SELECT \* FROM count\_number");

while($row=mysqli\_fetch\_assoc($find\_count))

{

$current\_count=$row['count1'];

$new\_count=$current\_count+1;

try{

$sql= "UPDATE count\_number SET count1 = '$new\_count' where id=1";

$update\_count=mysqli\_query ($conn,$sql);

echo $new\_count;

}

catch(Exception $e){

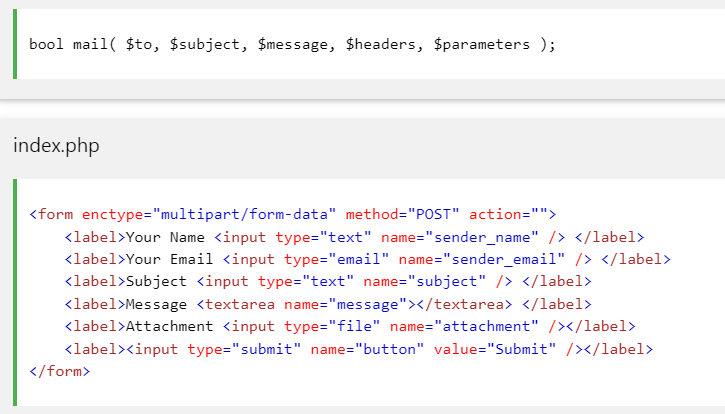
echo 'error;'.$e->getMessage();

}

}

?>

# Mail with attachment



<?php

if($\_POST['button'] && isset($\_FILES['attachment']))

{

    $from\_email      = 'sender@abc.com'; //from mail, sender email addrress

    $recipient\_email = 'recipient@xyz.com'; //recipient email addrress

    //Load POST data from HTML form

    $sender\_name = $\_POST["sender\_name"] //sender name

    $reply\_to\_email = $\_POST["sender\_email"] //sender email, it will be used in "reply-to" header

    $subject     = $\_POST["subject"] //subject for the email

    $message     = $\_POST["message"] //body of the email

    /\*Always remember to validate the form fields like this

    if(strlen($sender\_name)<1)

    {

        die('Name is too short or empty!');

    }

    \*/

    //Get uploaded file data using $\_FILES array

    $tmp\_name = $\_FILES['my\_file']['tmp\_name']; // get the temporary file name of the file on the server

    $name    = $\_FILES['my\_file']['name']; // get the name of the file

    $size    = $\_FILES['my\_file']['size']; // get size of the file for size validation

    $type    = $\_FILES['my\_file']['type']; // get type of the file

    $error   = $\_FILES['my\_file']['error']; // get the error (if any)

    //validate form field for attaching the file

    if($file\_error > 0)

    {

        die('Upload error or No files uploaded');

    }

    //read from the uploaded file & base64\_encode content

    $handle = fopen($tmp\_name, "r"); // set the file handle only for reading the file

    $content = fread($handle, $size); // reading the file

    fclose($handle);                 // close upon completion

    $encoded\_content = chunk\_split(base64\_encode($content));

    $boundary = md5("random"); // define boundary with a md5 hashed value

    //header

    $headers = "MIME-Version: 1.0\r\n"; // Defining the MIME version

    $headers .= "From:".$from\_email."\r\n"; // Sender Email

    $headers .= "Reply-To: ".$reply\_to\_email."\r\n"; // Email addrress to reach back

    $headers .= "Content-Type: multipart/mixed;\r\n"; // Defining Content-Type

    $headers .= "boundary = $boundary\r\n"; //Defining the Boundary

    //plain text

    $body = "--$boundary\r\n";

    $body .= "Content-Type: text/plain; charset=ISO-8859-1\r\n";

    $body .= "Content-Transfer-Encoding: base64\r\n\r\n";

    $body .= chunk\_split(base64\_encode($message));

    //attachment

    $body .= "--$boundary\r\n";

    $body .="Content-Type: $file\_type; name=".$file\_name."\r\n";

    $body .="Content-Disposition: attachment; filename=".$file\_name."\r\n";

    $body .="Content-Transfer-Encoding: base64\r\n";

    $body .="X-Attachment-Id: ".rand(1000, 99999)."\r\n\r\n";

    $body .= $encoded\_content; // Attaching the encoded file with email

    $sentMailResult = mail($recipient\_email, $subject, $body, $headers);

    if($sentMailResult )

    {

    echo "File Sent Successfully.";

    unlink($name); // delete the file after attachment sent.

    }

    else

    {

    die("Sorry but the email could not be sent.

                    Please go back and try again!");

    }

}

?>

**Pagination**



<html>

  <head>

    <title>Pagination</title>

    <link rel="stylesheet"

    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

    <style>

    table {

        border-collapse: collapse;

    }

        .inline{

            display: inline-block;

            float: right;

            margin: 20px 0px;

        }

        input, button{

            height: 34px;

        }

    .pagination {

        display: inline-block;

    }

    .pagination a {

        font-weight:bold;

        font-size:18px;

        color: black;

        float: left;

        padding: 8px 16px;

        text-decoration: none;

        border:1px solid black;

    }

    .pagination a.active {

            background-color: pink;

    }

    .pagination a:hover:not(.active) {

        background-color: skyblue;

    }

        </style>

  </head>

  <body>

  <center>

    <?php

    // Import the file where we defined the connection to Database.

        require\_once "connection.php";

        $per\_page\_record = 4;  // Number of entries to show in a page.

        // Look for a GET variable page if not found default is 1.

        if (isset($\_GET["page"])) {

            $page  = $\_GET["page"];

        }

        else {

          $page=1;

        }

        $start\_from = ($page-1) \* $per\_page\_record;

        $query = "SELECT \* FROM student LIMIT $start\_from, $per\_page\_record";

        $rs\_result = mysqli\_query ($conn, $query);

    ?>

    <div class="container">

      <br>

      <div>

        <h1>Pagination Simple Example</h1>

        <p>This page demonstrates the basic

           Pagination using PHP and MySQL.

        </p>

        <table class="table table-striped table-condensed

                                          table-bordered">

          <thead>

            <tr>

              <th width="10%">ID</th>

              <th>Name</th>

              <th>College</th>

              <th>Score</th>

            </tr>

          </thead>

          <tbody>

    <?php

            while ($row = mysqli\_fetch\_array($rs\_result)) {

                  // Display each field of the records.

            ?>

            <tr>

             <td><?php echo $row["Rank"]; ?></td>

            <td><?php echo $row["Name"]; ?></td>

            <td><?php echo $row["College"]; ?></td>

            <td><?php echo $row["Score"]; ?></td>

            </tr>

            <?php

                };

            ?>

          </tbody>

        </table>

     <div class="pagination">

      <?php

        $query = "SELECT COUNT(\*) FROM student";

        $rs\_result = mysqli\_query($conn, $query);

        $row = mysqli\_fetch\_row($rs\_result);

        $total\_records = $row[0];

    echo "</br>";

        // Number of pages required.

        $total\_pages = ceil($total\_records / $per\_page\_record);

        $pagLink = "";

        if($page>=2){

            echo "<a href='index1.php?page=".($page-1)."'>  Prev </a>";

        }

        for ($i=1; $i<=$total\_pages; $i++) {

          if ($i == $page) {

              $pagLink .= "<a class = 'active' href='index1.php?page="

                                                .$i."'>".$i." </a>";

          }

          else  {

              $pagLink .= "<a href='index1.php?page=".$i."'>

                                                ".$i." </a>";

          }

        };

        echo $pagLink;

        if($page<$total\_pages){

            echo "<a href='index1.php?page=".($page+1)."'>  Next </a>";

        }

      ?>

      </div>

      <div class="inline">

      <input id="page" type="number" min="1" max="<?php echo $total\_pages?>"

      placeholder="<?php echo $page."/".$total\_pages; ?>" required>

      <button onClick="go2Page();">Go</button>

     </div>

    </div>

  </div>

</center>

  <script>

    function go2Page()

    {

        var page = document.getElementById("page").value;

        page = ((page><?php echo $total\_pages; ?>)?<?php echo $total\_pages; ?>:((page<1)?1:page));

        window.location.href = 'index1.php?page='+page;

    }

  </script>

  </body>

</html>

**PHP Pegination 2**

<?php

$con=mysqli\_connect('localhost','root','','youtube');

$per\_page=5;

$start=0;

$current\_page=1;

if(isset($\_GET['start'])){

    $start=$\_GET['start'];

    if($start<=0){

        $start=0;

        $current\_page=1;

    }else{

        $current\_page=$start;

        $start--;

        $start=$start\*$per\_page;

    }

}

$record=mysqli\_num\_rows(mysqli\_query($con,"select id,title from page"));

$pagi=ceil($record/$per\_page);

$sql="select id,title from page limit $start,$per\_page";

$res=mysqli\_query($con,$sql);

?>

<!DOCTYPE html>

<html lang="en">

<head>

  <title>Pagination Example</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" href="css/bootstrap.min.css">

  <script src="js/jquery.min.js"></script>

  <script src="js/bootstrap.min.js"></script>

  <style>

  .mt-100{margin-top:50px;}

  .mt-30{margin-top:30px;}

  .mb-30{margin-bottom:30px;}

  </style>

</head>

<body>

<div class="container mt-100">

  <h2 class="mb-30">Pagination Example</h2>

  <ul class="list-group">

    <?php

    if(mysqli\_num\_rows($res)>0){

    while($row=mysqli\_fetch\_assoc($res)){?>

        <li class="list-group-item"><?php echo $row['title']?></li>

    <?php } } else {?>

    No records

    <?php } ?>

  </ul>

  <ul class="pagination mt-30">

    <?php

    for($i=1;$i<=$pagi;$i++){

    $class='';

    if($current\_page==$i){

        ?><li class="page-item active"><a class="page-link" href="javascript:void(0)"><?php echo $i?></a></li><?php

    }else{

    ?>

        <li class="page-item"><a class="page-link" href="?start=<?php echo $i?>"><?php echo $i?></a></li>

    <?php

    }

    ?>

    <?php } ?>

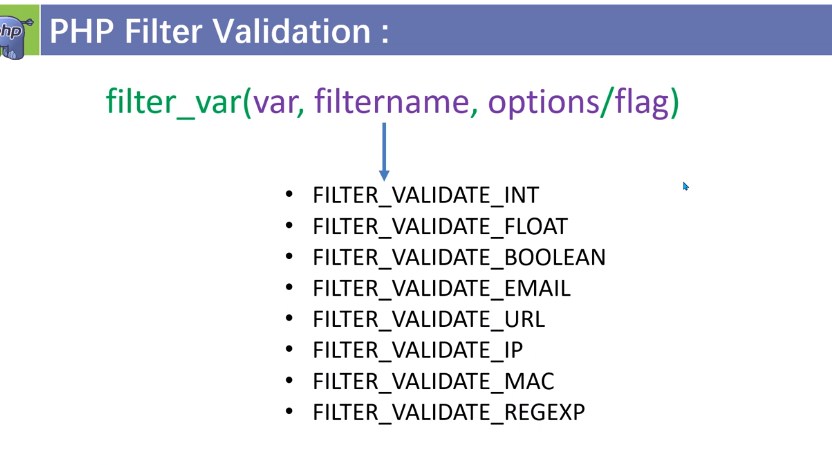
  </ul>

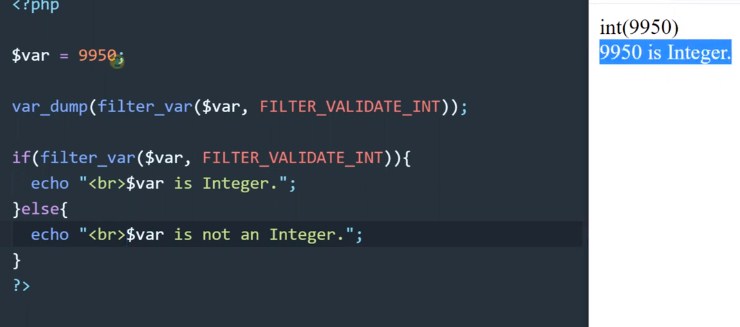
</div>

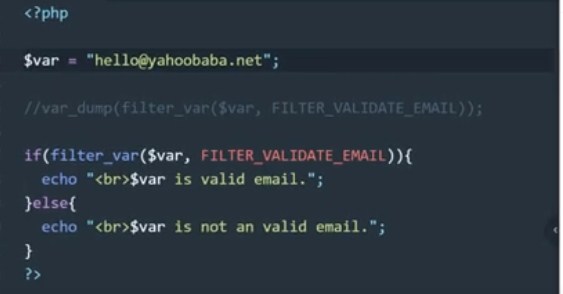
</body>

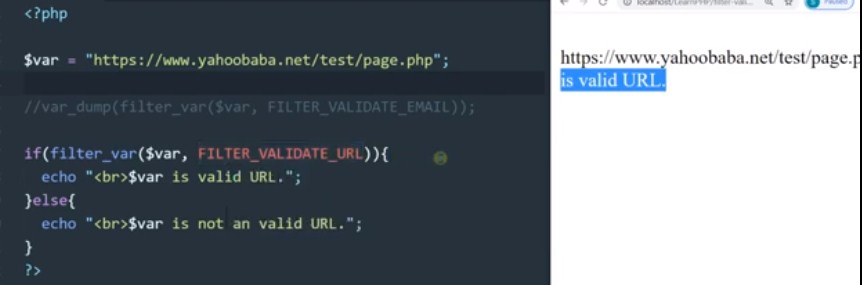
</html>

**Validation PHP function**

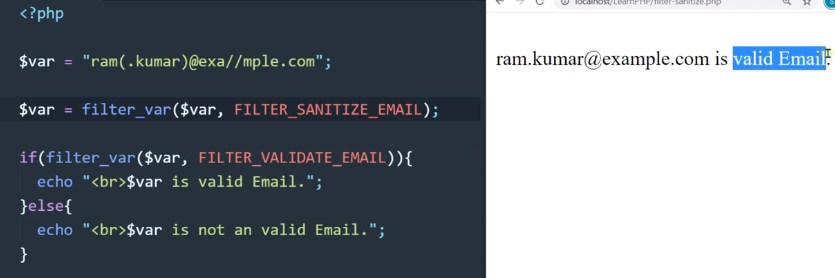
****

****

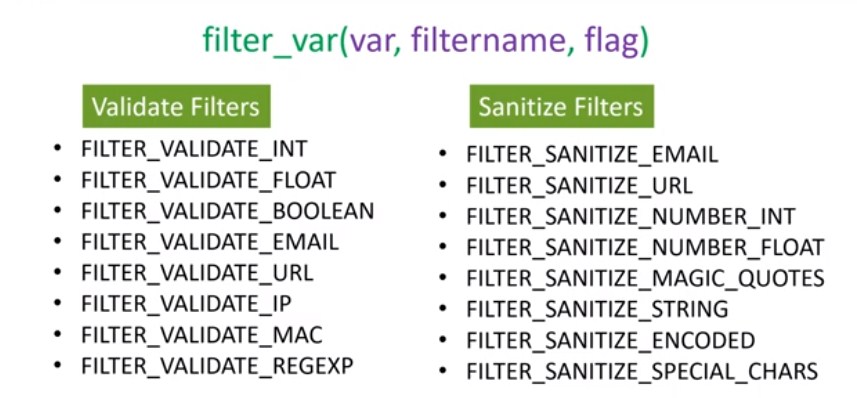
****

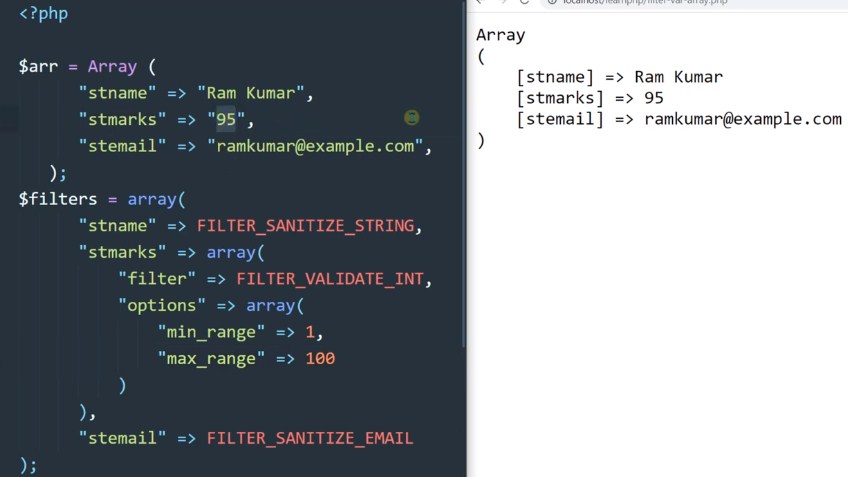
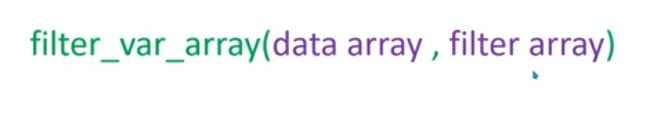
****

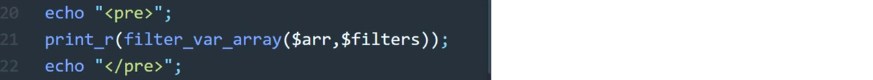
****

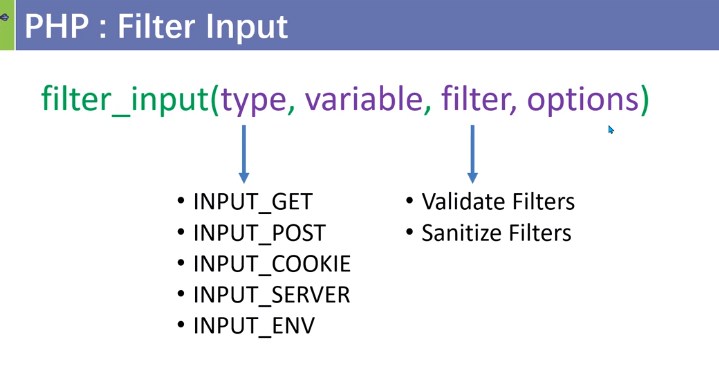
****

****

****

****

****

****

****

**Validation PHP**

<?php

// Functions to filter user inputs

function filterName($field){

    // Sanitize user name

    $field = filter\_var(trim($field), FILTER\_SANITIZE\_STRING);

    // Validate user name

    if(filter\_var($field, FILTER\_VALIDATE\_REGEXP, array("options"=>array("regexp"=>"/^[a-zA-Z\s]+$/")))){

        return $field;

    } else{

        return FALSE;

    }

}

function filterEmail($field){

    // Sanitize e-mail address

    $field = filter\_var(trim($field), FILTER\_SANITIZE\_EMAIL);

    // Validate e-mail address

    if(filter\_var($field, FILTER\_VALIDATE\_EMAIL)){

        return $field;

    } else{

        return FALSE;

    }

}

function filterString($field){

    // Sanitize string

    $field = filter\_var(trim($field), FILTER\_SANITIZE\_STRING);

    if(!empty($field)){

        return $field;

    } else{

        return FALSE;

    }

}

// Define variables and initialize with empty values

$nameErr = $emailErr = $messageErr = "";

$name = $email = $subject = $message = "";

// Processing form data when form is submitted

if($\_SERVER["REQUEST\_METHOD"] == "POST"){

    // Validate user name

    if(empty($\_POST["name"])){

        $nameErr = "Please enter your name.";

    } else{

        $name = filterName($\_POST["name"]);

        if($name == FALSE){

            $nameErr = "Please enter a valid name.";

        }

    }

    // Validate email address

    if(empty($\_POST["email"])){

        $emailErr = "Please enter your email address.";

    } else{

        $email = filterEmail($\_POST["email"]);

        if($email == FALSE){

            $emailErr = "Please enter a valid email address.";

        }

    }

    // Validate message subject

    if(empty($\_POST["subject"])){

        $subject = "";

    } else{

        $subject = filterString($\_POST["subject"]);

    }

    // Validate user comment

    if(empty($\_POST["message"])){

        $messageErr = "Please enter your comment.";

    } else{

        $message = filterString($\_POST["message"]);

        if($message == FALSE){

            $messageErr = "Please enter a valid comment.";

        }

    }

    // Check input errors before sending email

    if(empty($nameErr) && empty($emailErr) && empty($messageErr)){

        // Recipient email address

        $to = 'webmaster@example.com';

        // Create email headers

        $headers = 'From: '. $email . "\r\n" .

        'Reply-To: '. $email . "\r\n" .

        'X-Mailer: PHP/' . phpversion();

        // Sending email

        if(mail($to, $subject, $message, $headers)){

            echo '<p class="success">Your message has been sent successfully!</p>';

        } else{

            echo '<p class="error">Unable to send email. Please try again!</p>';

        }

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Contact Form</title>

    <style type="text/css">

        .error{ color: red; }

        .success{ color: green; }

    </style>

</head>

<body>

    <h2>Contact Us</h2>

    <p>Please fill in this form and send us.</p>

    <form action="contact.php" method="post">

        <p>

            <label for="inputName">Name:<sup>\*</sup></label>

            <input type="text" name="name" id="inputName" value="<?php echo $name; ?>">

            <span class="error"><?php echo $nameErr; ?></span>

        </p>

        <p>

            <label for="inputEmail">Email:<sup>\*</sup></label>

            <input type="text" name="email" id="inputEmail" value="<?php echo $email; ?>">

            <span class="error"><?php echo $emailErr; ?></span>

        </p>

        <p>

            <label for="inputSubject">Subject:</label>

            <input type="text" name="subject" id="inputSubject" value="<?php echo $subject; ?>">

        </p>

        <p>

            <label for="inputComment">Message:<sup>\*</sup></label>

            <textarea name="message" id="inputComment" rows="5" cols="30"><?php echo $message; ?></textarea>

            <span class="error"><?php echo $messageErr; ?></span>

        </p>

        <input type="submit" value="Send">

        <input type="reset" value="Reset">

    </form>

</body>

</html>

**Another Example**

<!DOCTYPE HTML>

<html>

<head>

<style>

.error {color: #FF0000;}

</style>

</head>

<body>

<?php

/\* define variables and set to empty values \*/

$nameErr = $emailError = $mobileError ="";

$name = $email = $mobile = "";

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (empty($\_POST["name"])) {

$nameErr = "Name is required";

}

else {

$name = test\_input($\_POST["name"]);

/\* check if name only contains letters and whitespace \*/

if (!preg\_match("/^[a-zA-Z ]\*$/",$name)) {

$nameErr = "Only letters and white space allowed";

}

}

if (empty($\_POST["name"])) {

$mobileError = "Name is required";

} else {

$mobile = test\_input($\_POST["mobile"]);

/\* check if name only contains letters and whitespace \*/

if (!preg\_match('/^[0-9]{10}+$/', $mobile)) {

$mobileError = "10 digit Number";

}

}

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

$emailError = "Email is required";

} else {

$email = test\_input($\_POST["email"]);

/\* check if e-mail address is well-formed \*/

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$emailError = "Invalid email format";

}

}

}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

Name: <input type="text" name="name" value="<?php echo $name;?>">

<span class="error">\* <?php echo $nameErr;?></span>

<br><br>

Mobile: <input type="text" name="mobile" value="<?php echo $mobile;?>">

<span class="error">\* <?php echo $mobileError;?></span>

<br><br>

E-mail: <input type="text" name="email" value="<?php echo $email;?>">

<span class="error">\* <?php echo $emailError;?></span>

<br><br>

<input type="submit" name="submit" value="Submit">

</form>

</body>

</html>

**Login and signup with facebook**

<!DOCTYPE html>

<html>

<title>HTML Tutorial</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script> <body>

<!-- Display login status -->

<div id="status"></div>

<!-- Facebook login or logout button -->

<a href="javascript:void(0);" onclick="fbLogin();" id="fbLink"><img src="images/fb-login-btn.png"/></a>

<!-- Display user's profile info -->

<div class="ac-data" id="userData"></div>

<script>

window.fbAsyncInit = function() {

    /\* FB JavaScript SDK configuration and setup \*/

    FB.init({

      appId      : '264467978157588', /\* FB App ID \*/

      cookie     : true,  /\* enable cookies to allow the server to access the session \*/

      xfbml      : true,  /\* parse social plugins on this page \*/

      version    : 'v3.2' /\* use graph api version 2.8 \*/

    });

    /\* Check whether the user already logged in \*/

    FB.getLoginStatus(function(response) {

        if (response.status === 'connected') {

            /\* display user data \*/

            getFbUserData();

        }

    });

};

/\* Load the JavaScript SDK asynchronously \*/

(function(d, s, id) {

    var js, fjs = d.getElementsByTagName(s)[0];

    if (d.getElementById(id)) return;

    js = d.createElement(s); js.id = id;

    js.src = "/\* connect.facebook.net/en\_US/sdk.js"; \*/

    fjs.parentNode.insertBefore(js, fjs);

}(document, 'script', 'facebook-jssdk'));

/\* Facebook login with JavaScript SDK \*/

function fbLogin() {

    FB.login(function (response) {

        if (response.authResponse) {

            /\* Get and display the user profile data \*/

            getFbUserData();

        } else {

            document.getElementById('status').innerHTML = 'User cancelled login or did not fully authorize.';

        }

    }, {scope: 'email'});

}

/\* Fetch the user profile data from facebook \*/

function getFbUserData(){

    FB.api('/me', {locale: 'en\_US', fields: 'id,first\_name,last\_name,email,link,gender,locale,picture'},

    function (response) {

        document.getElementById('fbLink').setAttribute("onclick","fbLogout()");

        document.getElementById('fbLink').innerHTML = 'Logout from Facebook';

        document.getElementById('status').innerHTML = '<p>Thanks for logging in, ' + response.first\_name + '!</p>';

        document.getElementById('userData').innerHTML = '<h2>Facebook Profile Details</h2><p><img src="'+response.picture.data.url+'"/></p><p><b>FB ID:</b> '+response.id+'</p><p><b>Name:</b> '+response.first\_name+' '+response.last\_name+'</p><p><b>Email:</b> '+response.email+'</p>';

    });

}

/\* Logout from facebook \*/

function fbLogout() {

    FB.logout(function() {

        document.getElementById('fbLink').setAttribute("onclick","fbLogin()");

        document.getElementById('fbLink').innerHTML = '<img src="images/fb-login-btn.png"/>';

        document.getElementById('userData').innerHTML = '';

        document.getElementById('status').innerHTML = '<p>You have successfully logout from Facebook.</p>';

    });

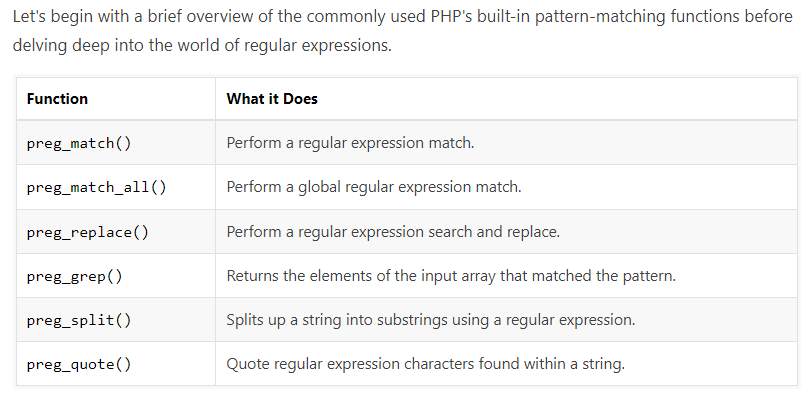
}

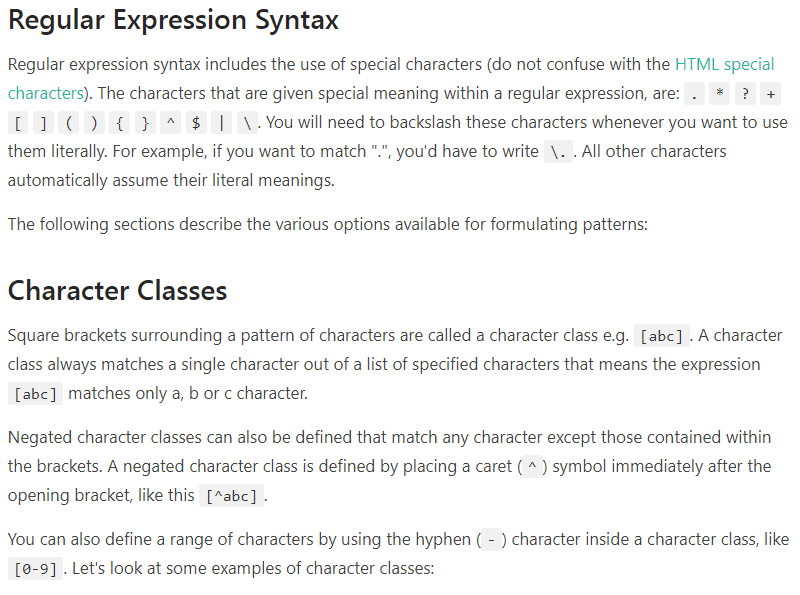
</script>

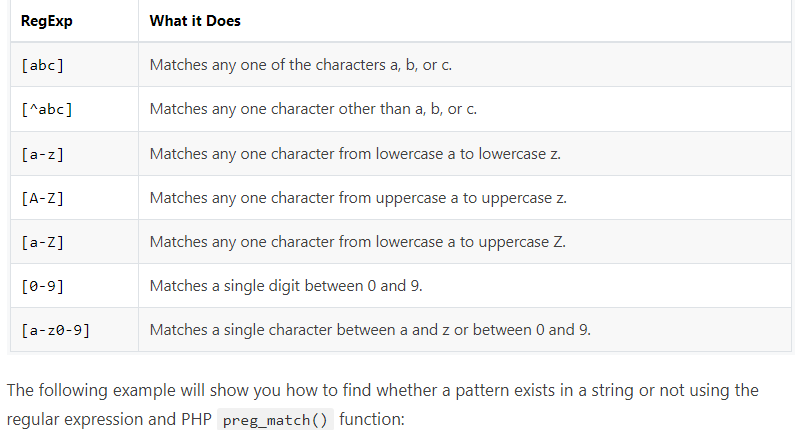
</body>

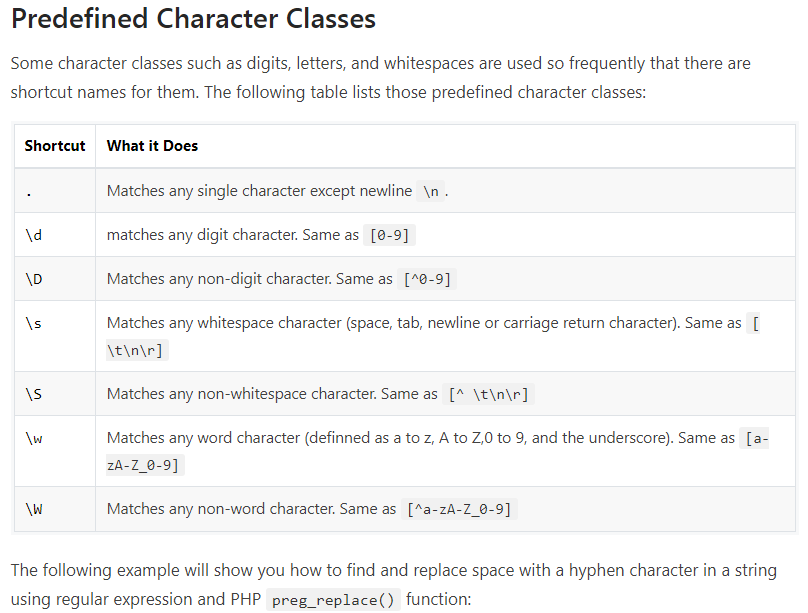
</html>

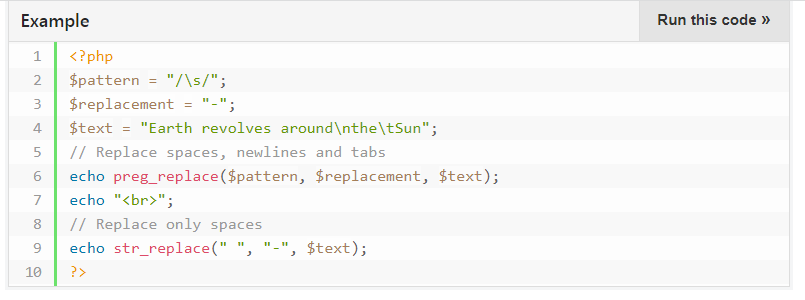
**Regular Expression**

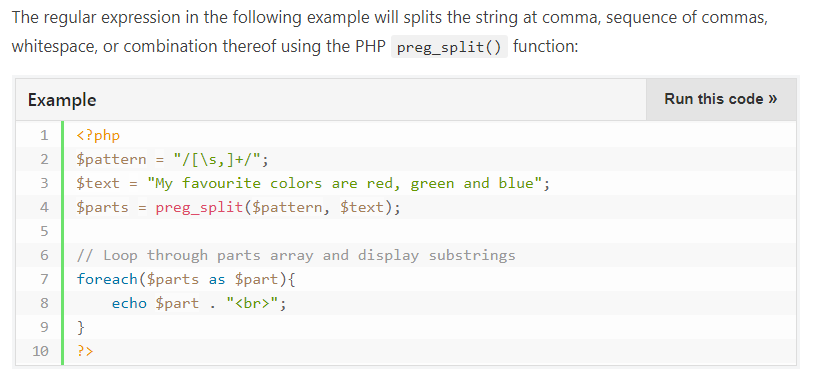
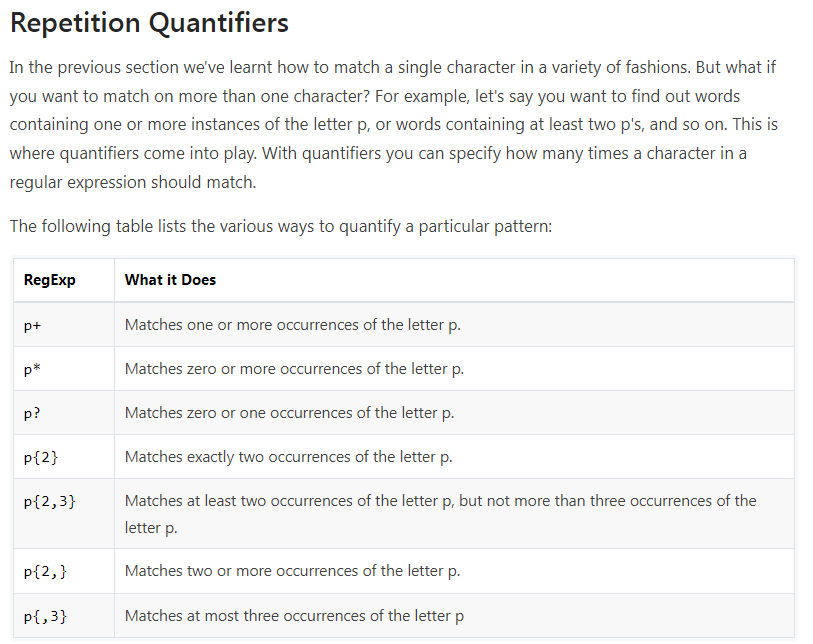


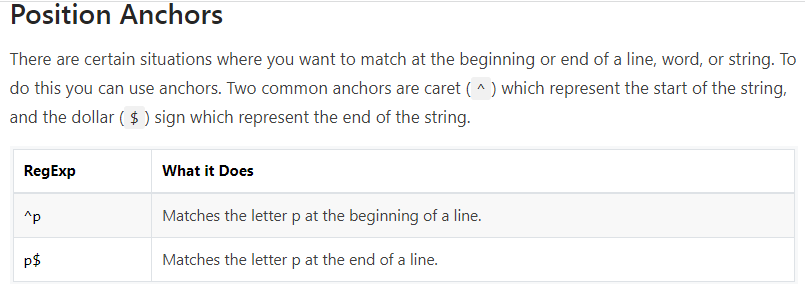




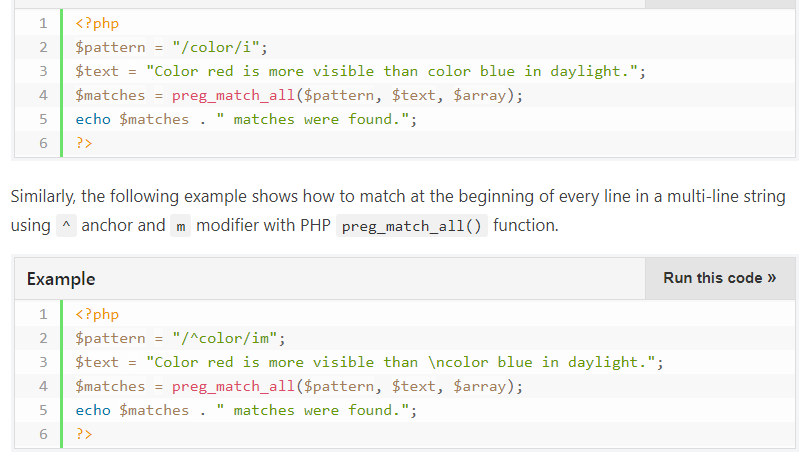
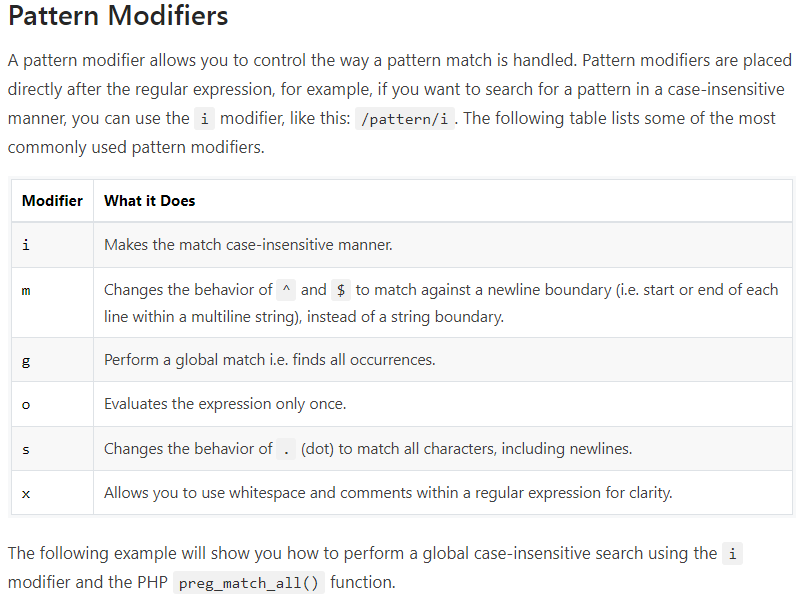


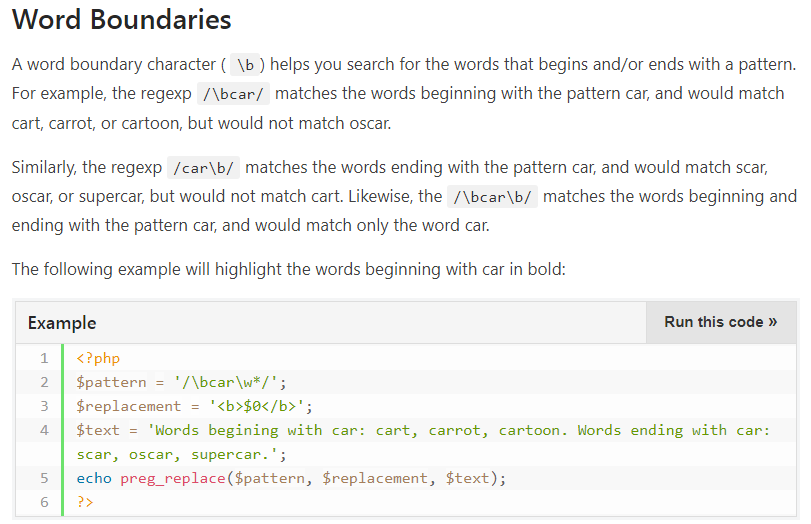


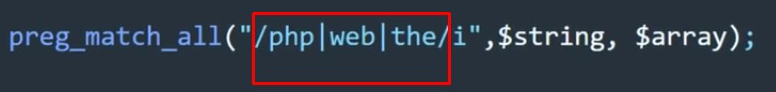


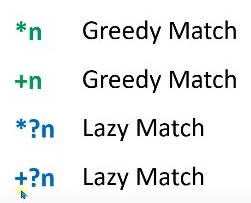
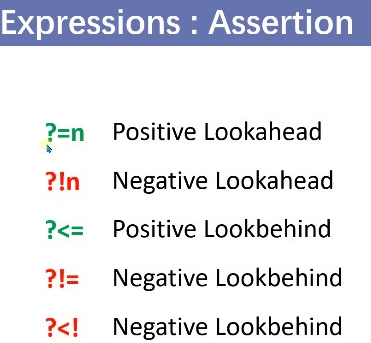
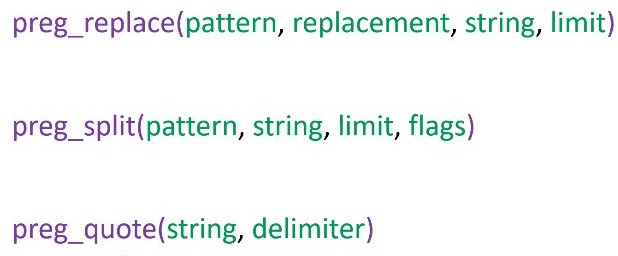
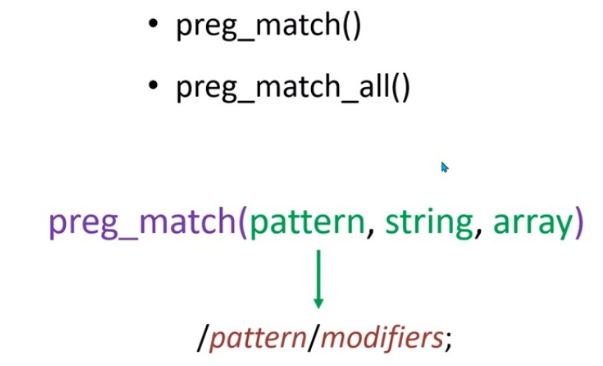


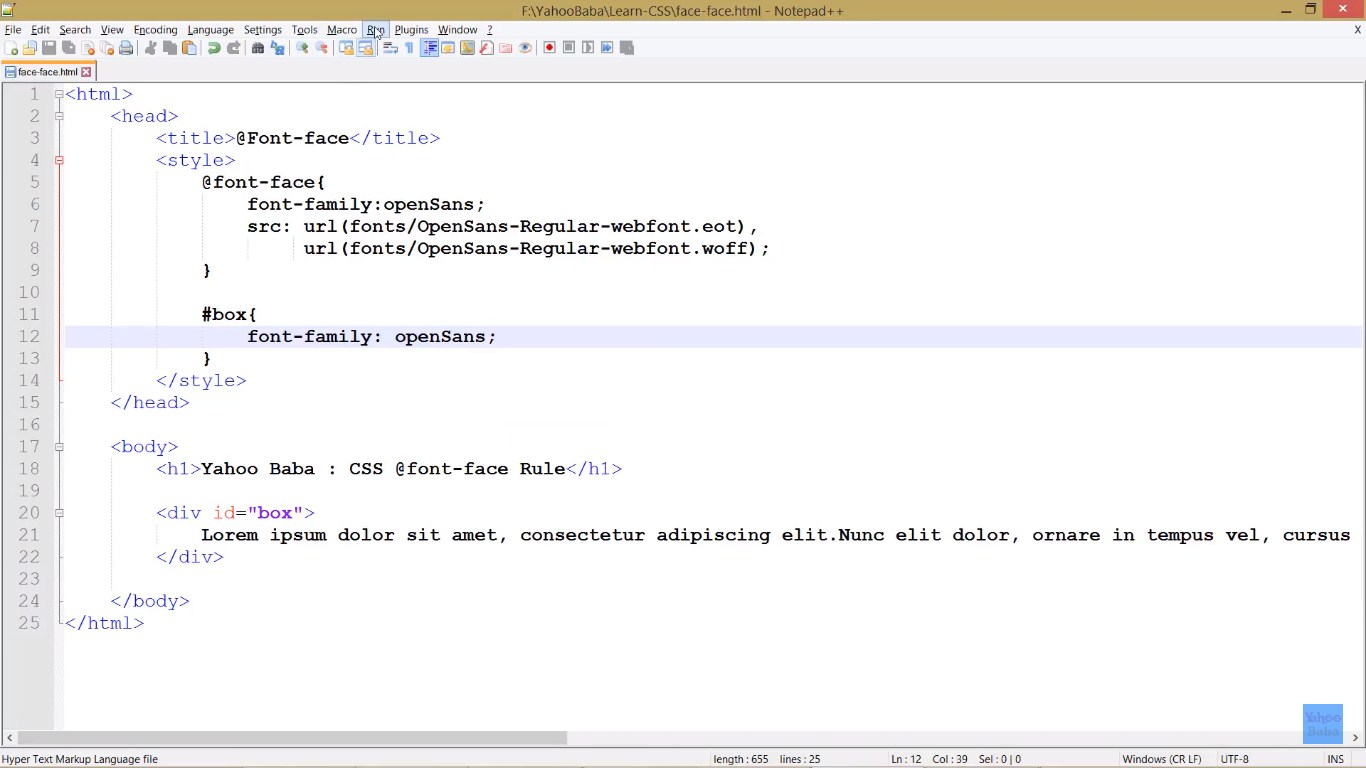
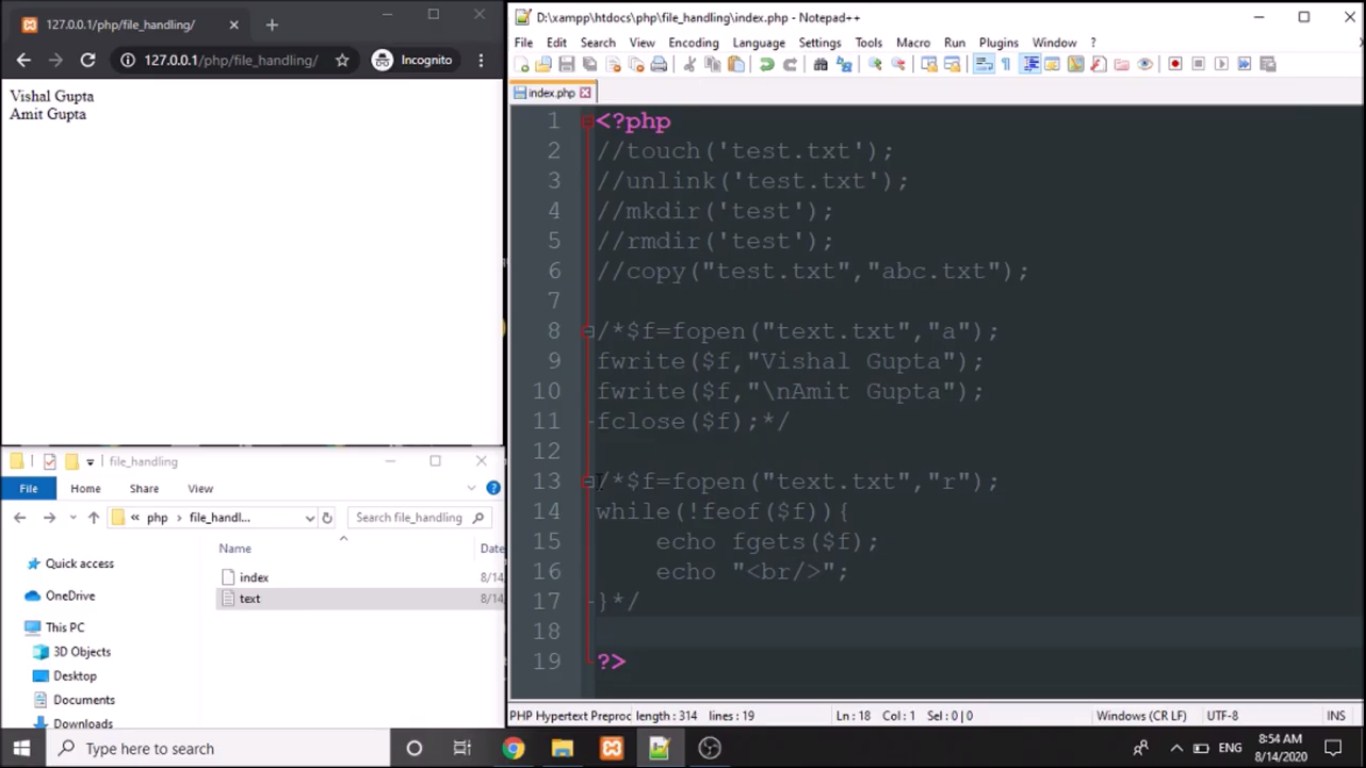
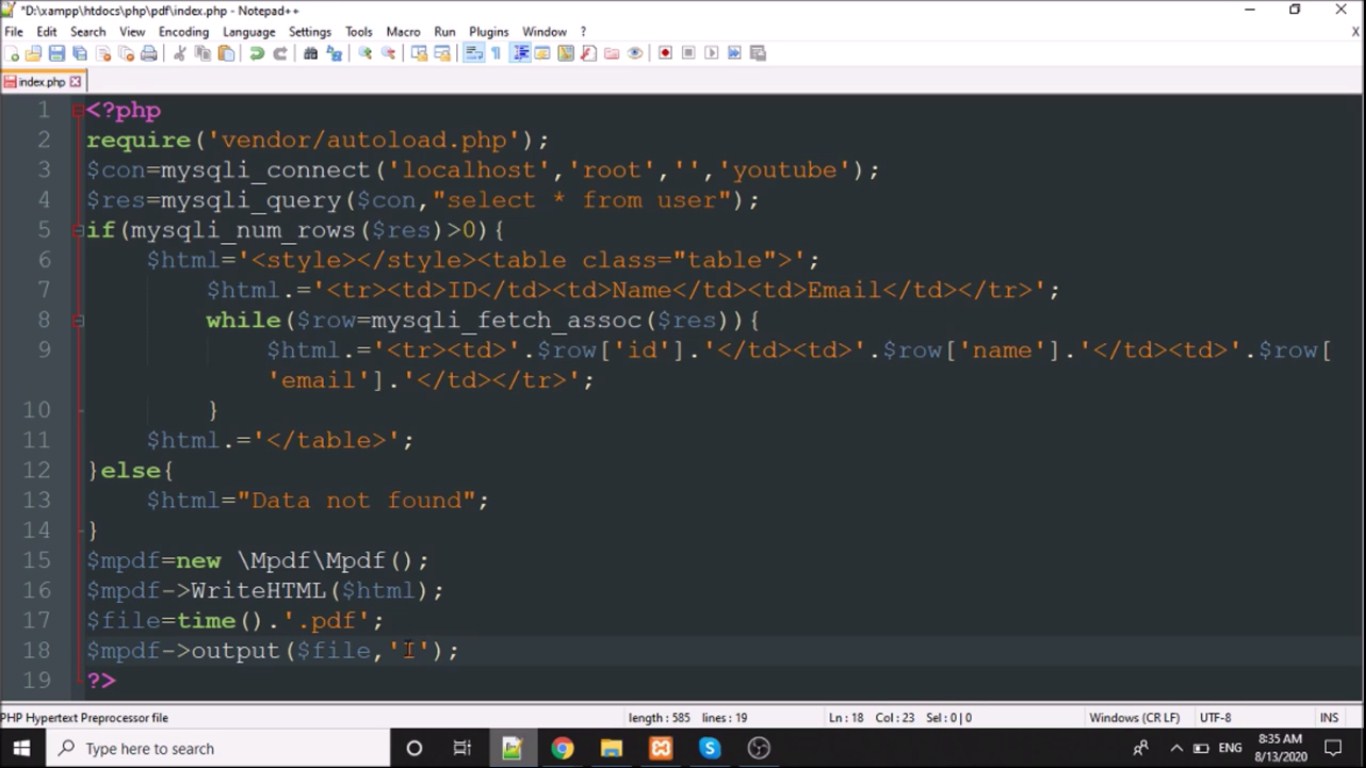
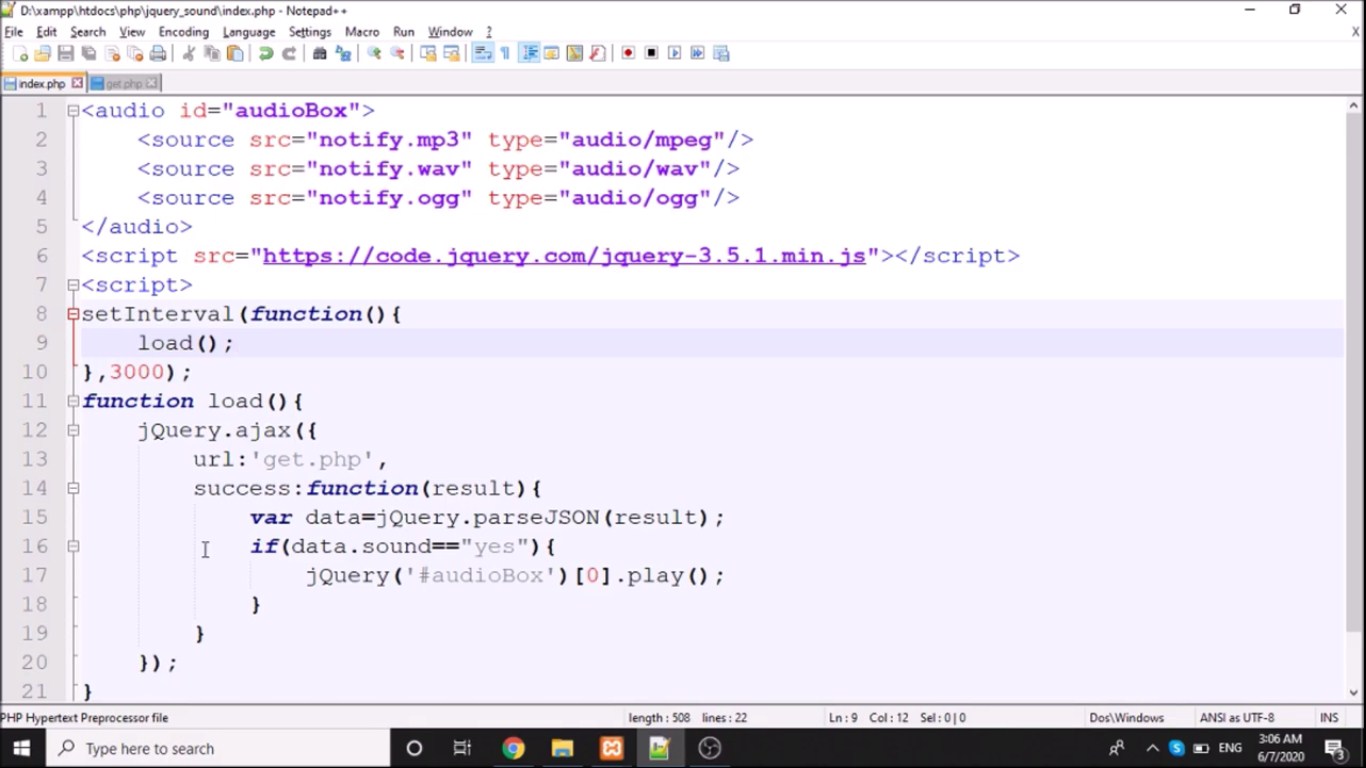
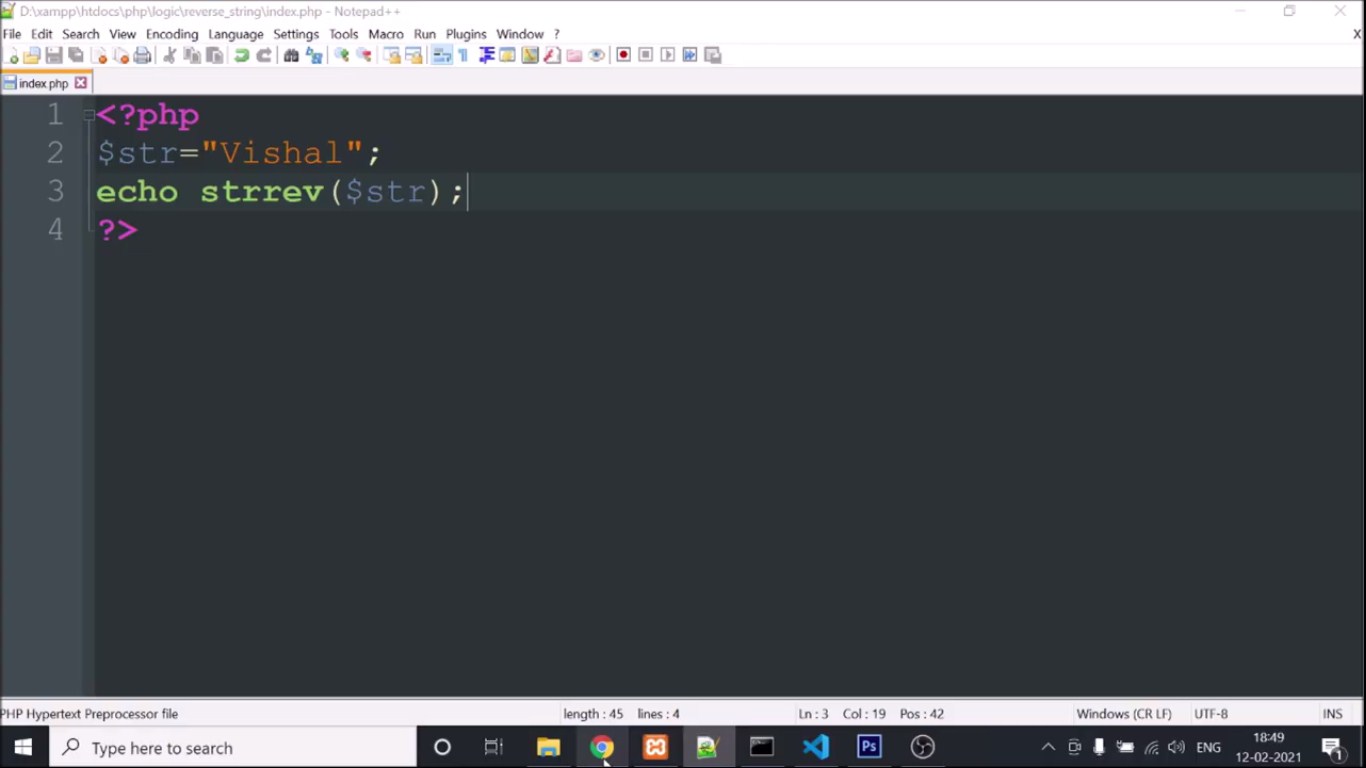
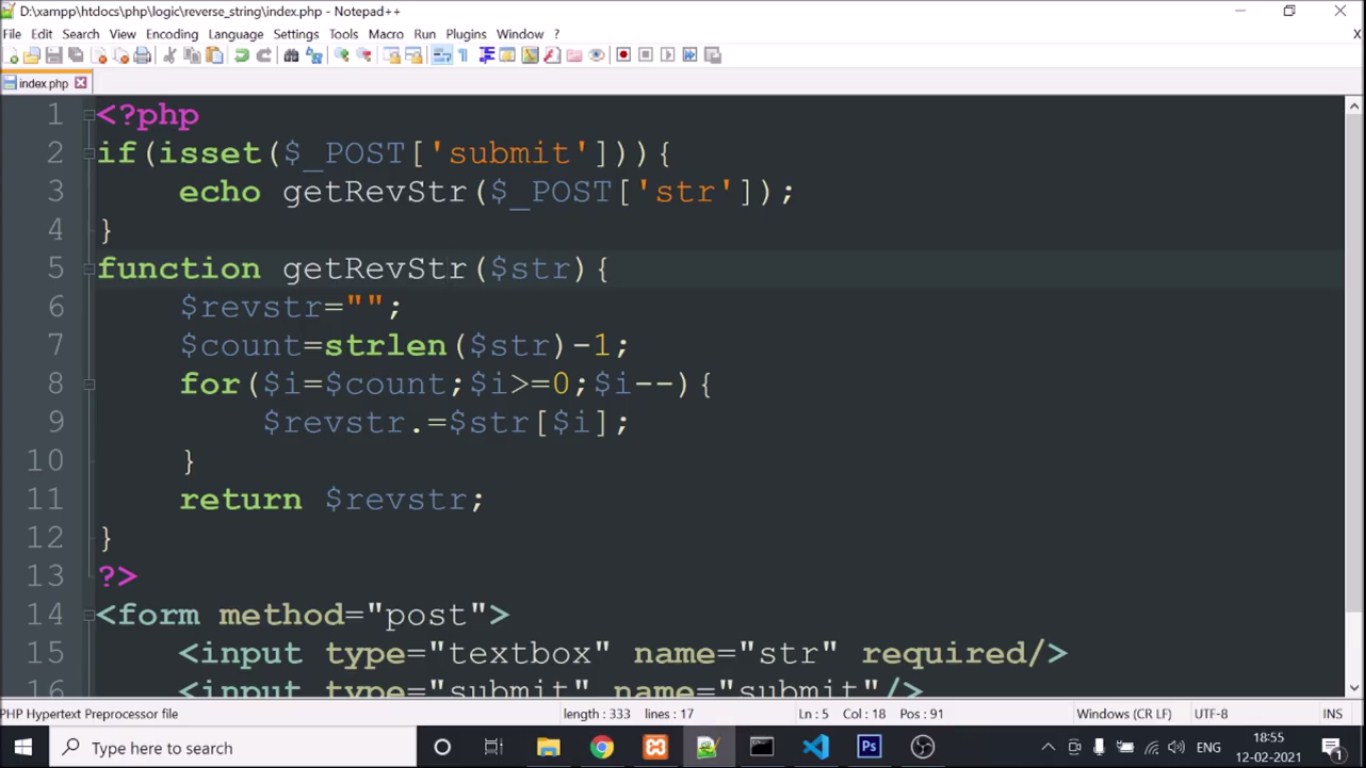


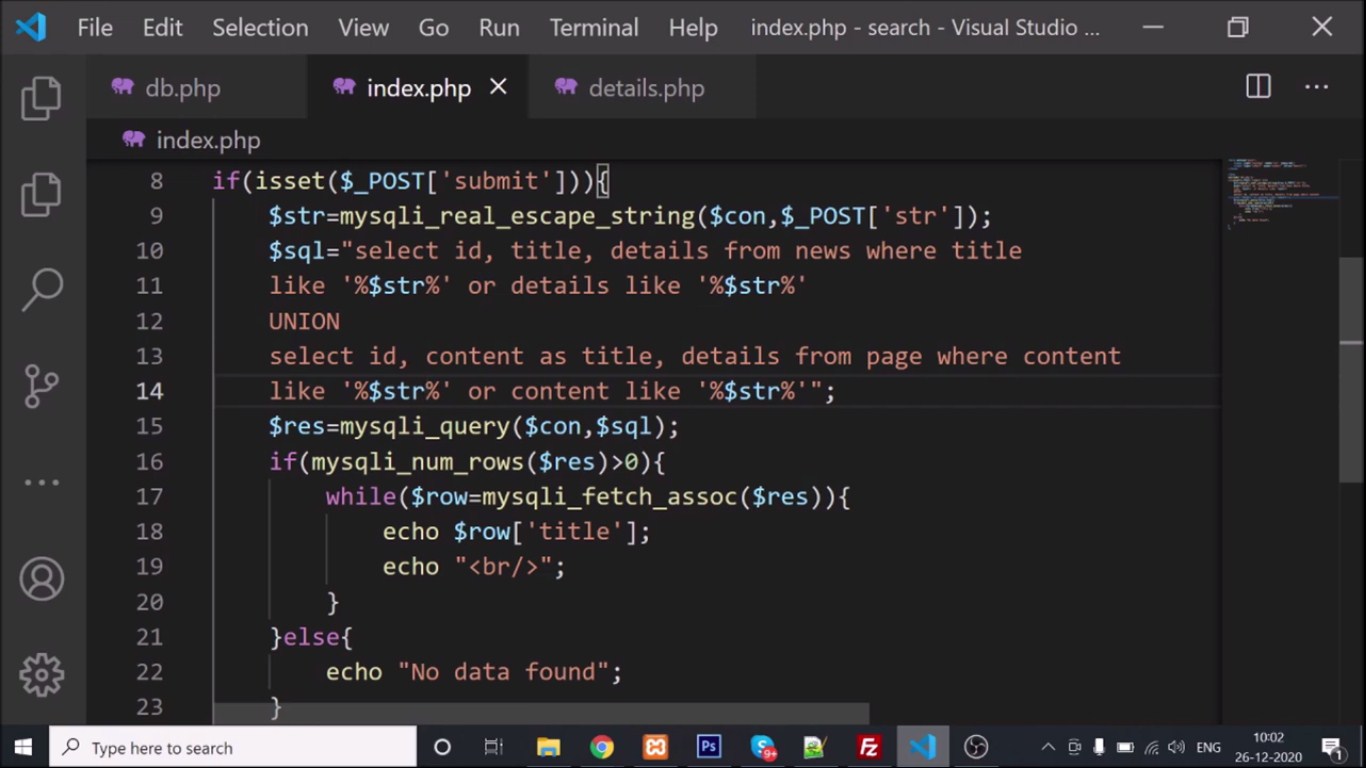


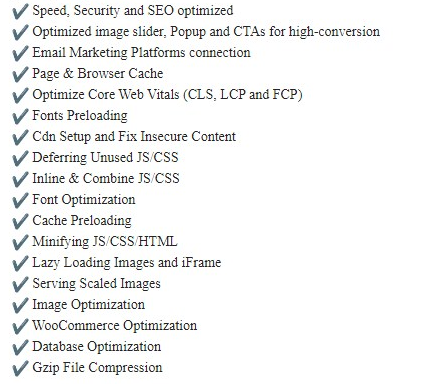


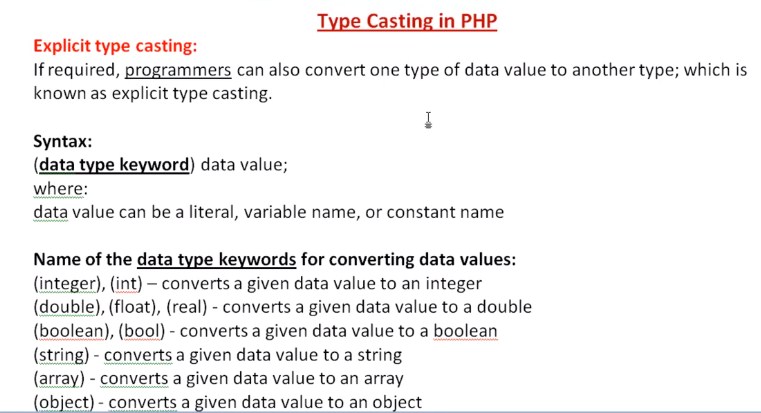
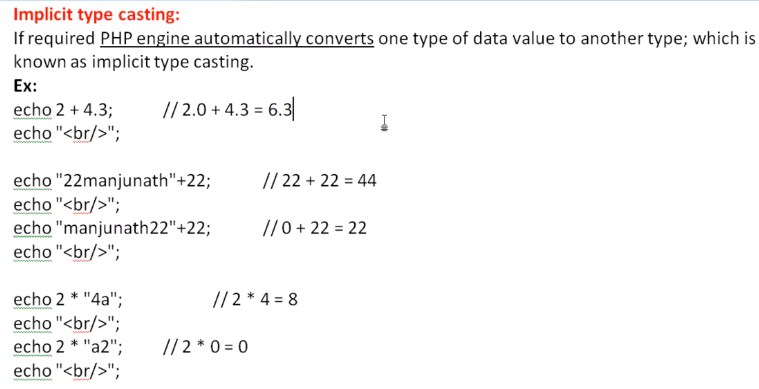
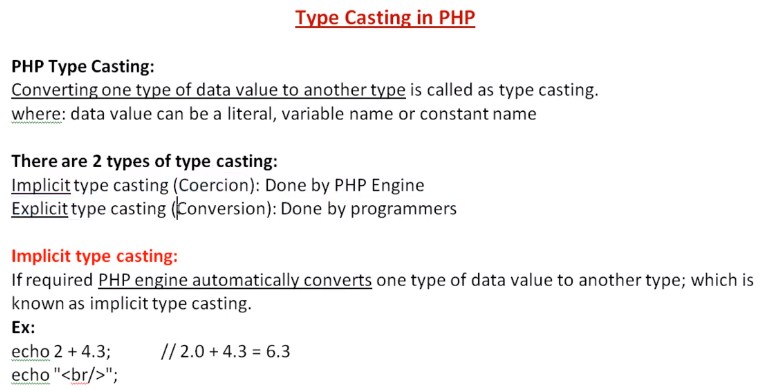


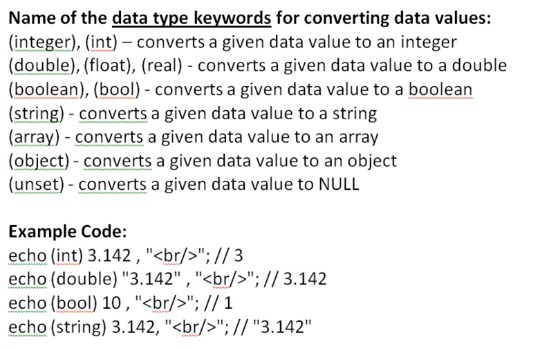


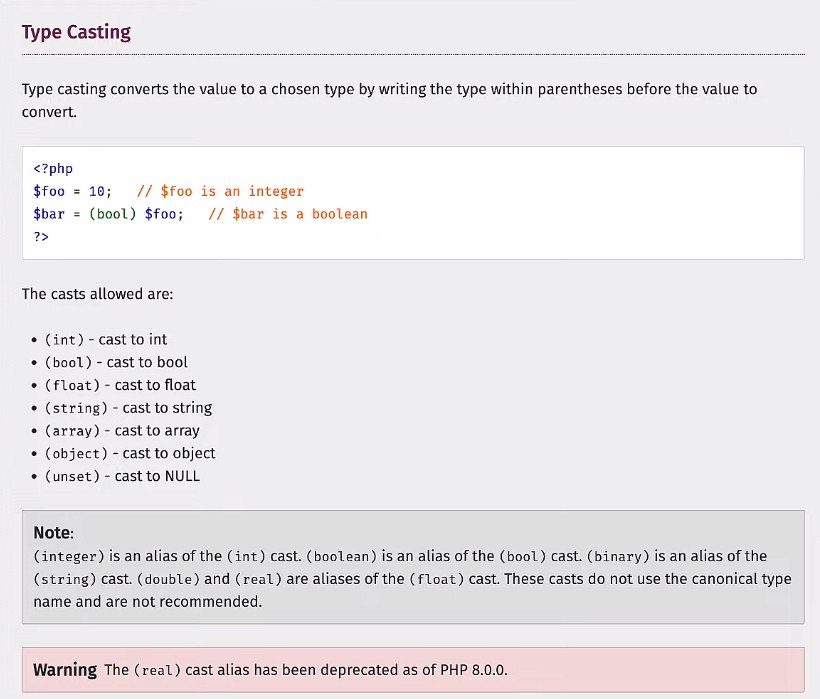


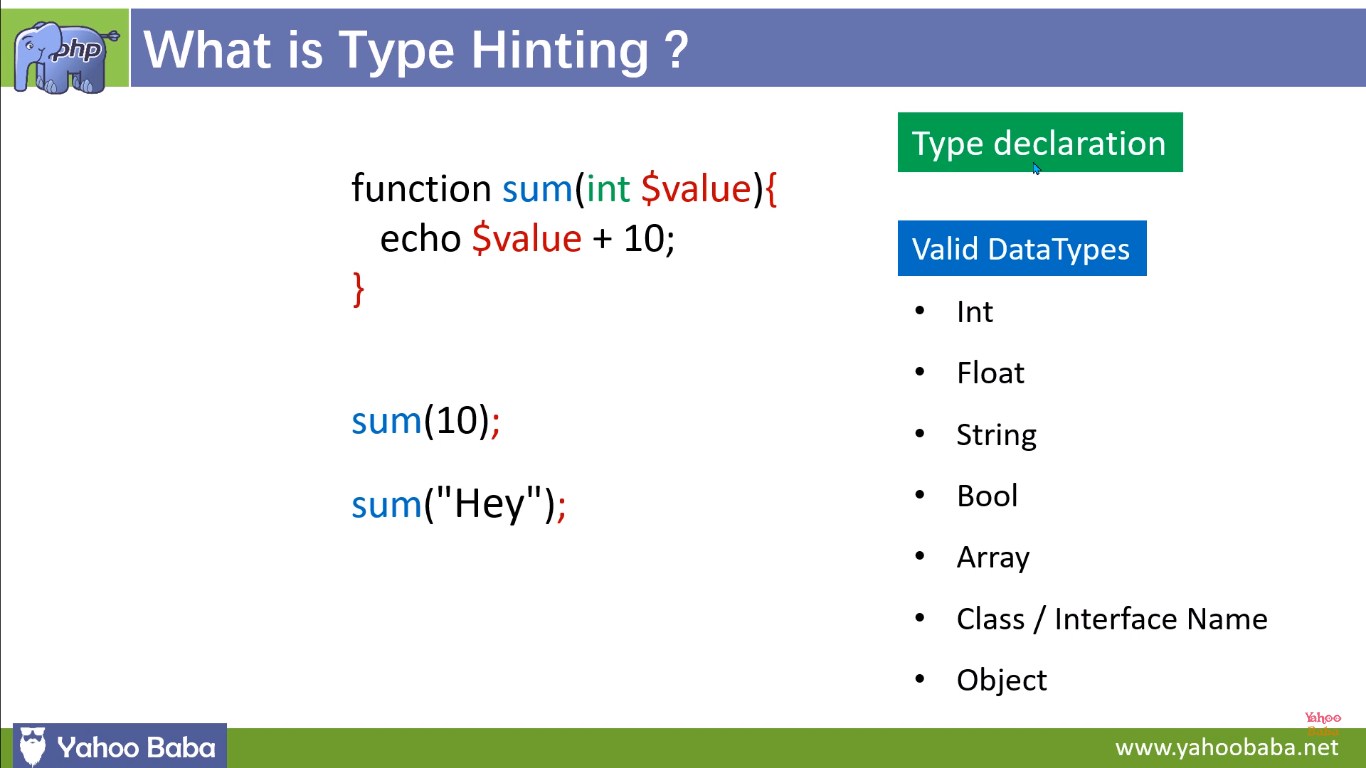


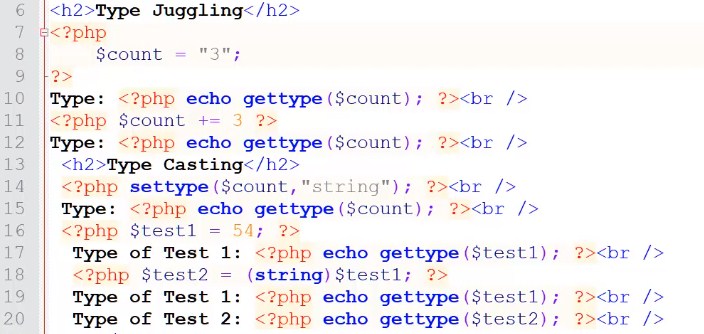


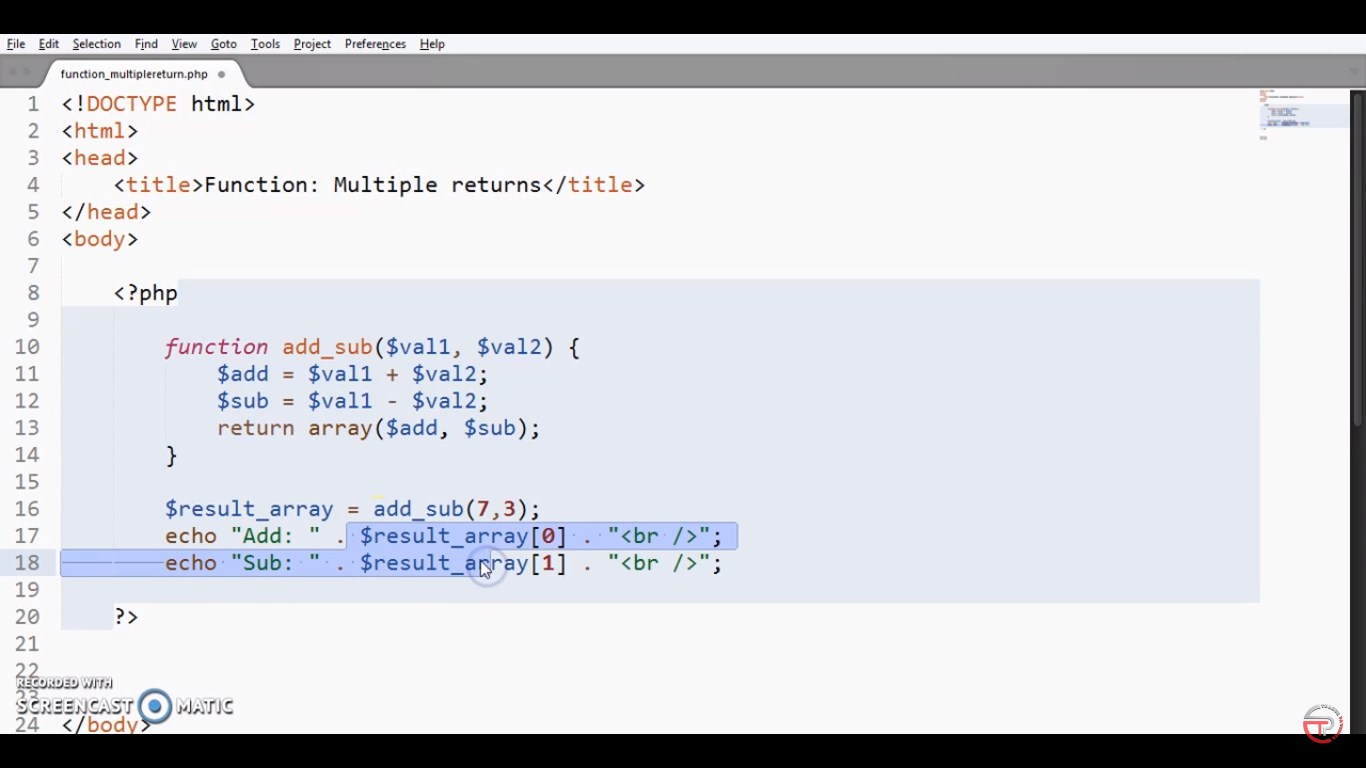
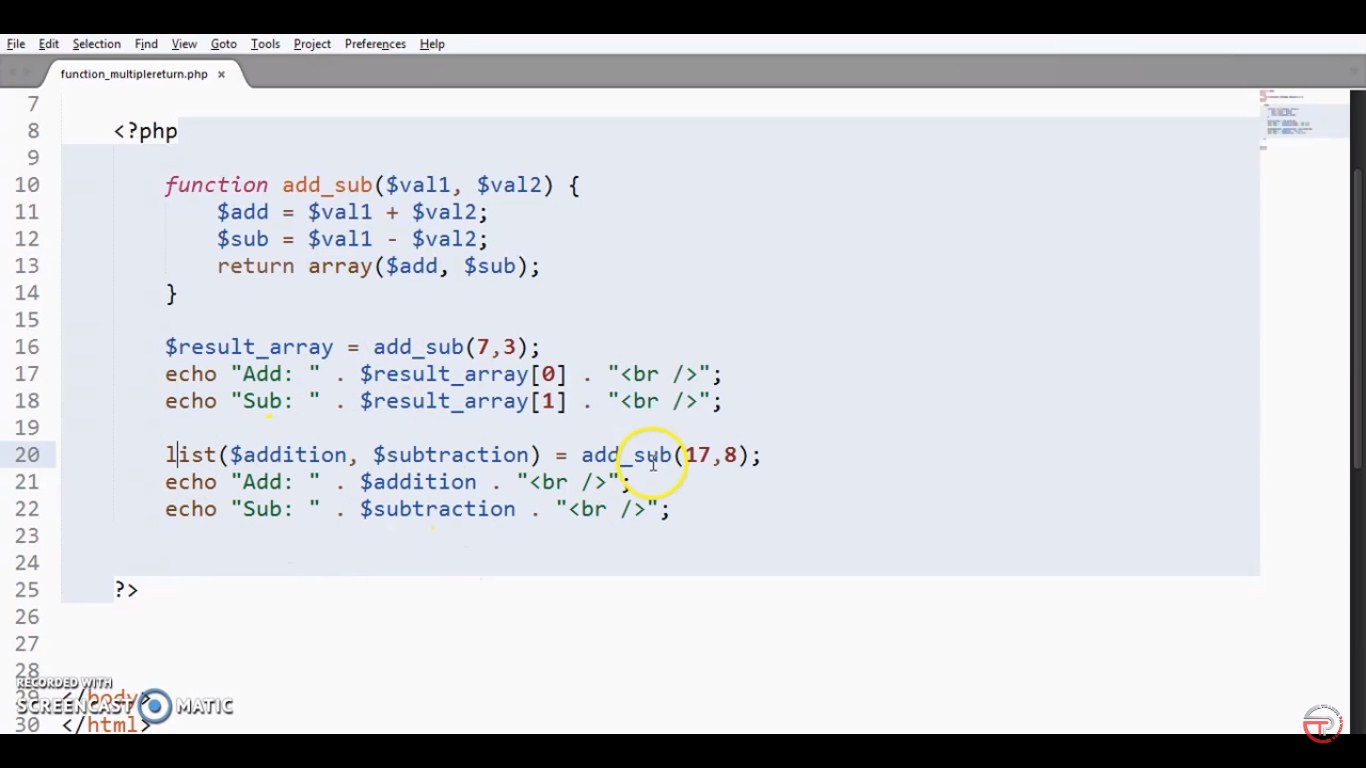
****

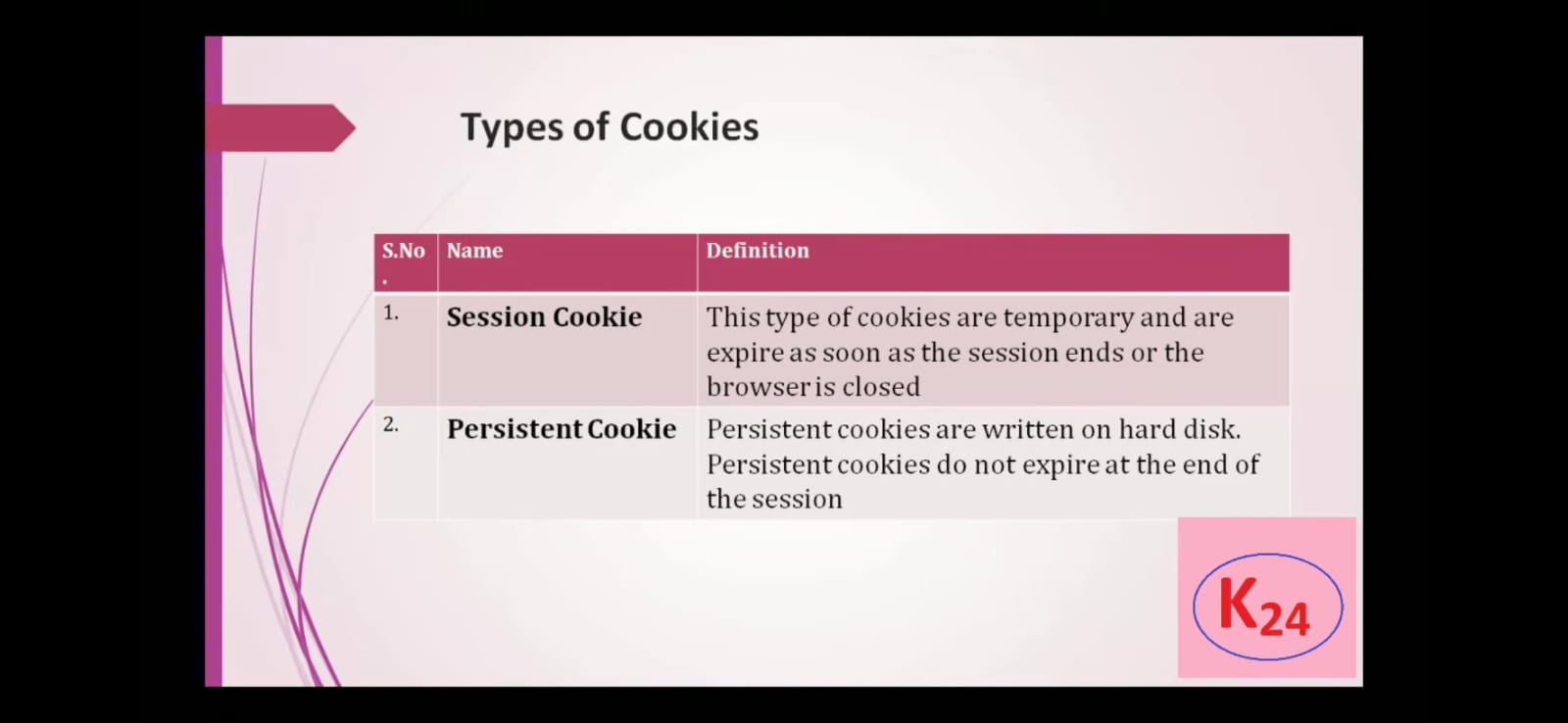
****

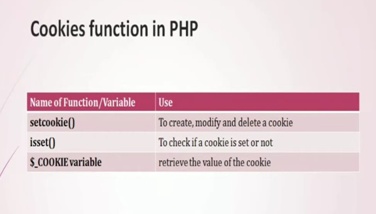












**IQ Question**

### include()****and**** require()

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

**How can we get the IP address of the client?**

$\_SERVER["REMOTE\_ADDR"];

## 

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

* **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.

### ****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.

### ****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.

## ****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.

### ****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';

}

**?>**

### ****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));

**?>**

## 

### ****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?****

**<?php**

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

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

}

**?>**

## ****How would you implement a class in the following scenario?****

**<?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;

}

}

}

**?>**

### ****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

### ****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.

### ****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.

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

## 

### ****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.

## ****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.

## ****What are PSRs? Choose 1 and briefly describe it.****

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

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

Developers should follow 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.

## ****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 managing the libraries.

### ****What is PHP most used for?****

**PHP** has a plethora of uses for developers and the ones mentioned below are some of the most widely used concepts that PHP offers:

* With PHP, it becomes very easy to provide restricted access to the required content of the website.
* It allows users to access individual cookies and set them as per requirement.
* Database manipulation operations, such as addition, deletion, and modification, can be done easily.
* Form handling, alongside features that involve file handling concepts and email integration, is used widely.
* The system module allows users to perform a variety of system functions such as open, read, write, etc.

### ****What is the meaning of PEAR in PHP?****

PEAR stands for PHP Extension and Application Repository. It is one of the frameworks and acting repositories that host all of the reusable PHP components. Alongside containing some of the PHP libraries, it also provides you with a simple interface in PHP to automatically install packages.

### ****How is a PHP script executed?****

PHP scripts can be easily executed from the command-line interface (CLI). The syntax is as follows:

php filename.php

Here, **filename**refers to the file that contains scripts. The extension **.php** is needed alongside the filename.

### ****What are the types of variables present in PHP?****

* **Array**: A named and ordered collection of data
* **Boolean**: A logical value (True or False)
* **Double**: Floating point numbers such as 5.1525
* **Integer**: Whole numbers without a floating point
* **Object**: An instance of classes, containing data and functions
* **NULL**: A special data type, supporting only the NULL data
* **Resource**: Special variables that hold references to external resources
* **String**: A sequence of characters such as, “Hello learners!”

### ****What are the variable-naming rules you should follow in PHP?****

* Variables can only begin with letters or underscores.
* Special characters such as +, %, -, &, etc. cannot be used.

### ****What are the main characteristics of a PHP variable?****

* Variables can be declared before the value assignment.
* A variable value assignment happens using the ‘=’ operator.
* Every variable in PHP is denoted with a $ (dollar) sign.
* The value of a variable depends on its latest assigned value.
* PHP variables are not intrinsic. There is no explicit declaration.

### ****What is NULL in PHP?****

NULL is a special data type in PHP used to denote the presence of only one value, NULL. You cannot assign any other value to it.

NULL is not case sensitive in PHP and can be declared in two ways as shown below:

$var = NULL:

Or

$var = null;

Both of the above syntaxes work fine in PHP.

### ****How are constants defined in PHP?****

Constants can be defined easily in PHP by making use of the **define()** function. This function is used to define and pull out the values of the constants easily.

Constants, as the name suggests, cannot be changed after being definition. They do not require the PHP syntax of starting with the conventional $ sign.

### ****What is the use of the constant() function in PHP?****

The constant() function is used to retrieve the values predefined in a constant variable. It is used especially when you do not know the name of the variable.

### ****Differentiate between variables and constants in PHP.****

|  |  |
| --- | --- |
| **Variable** | **Constant** |
| Variables can have changed paths | Constants cannot be changed |
| The default scope is the current access scope | Constants can be accessed throughout without any scoping rules |
| The $ assignment is used for definition | Constants are defined using the define() function |
| Compulsory usage of the $ sign at the start | No need for the $ sign for constants |

### ****What does the phrase ‘PHP escape’ mean?****

PHP escape is a mechanism that is used to tell the PHP parser that certain code elements are different from the PHP code. This provides the basic means to differentiate a piece of PHP code from the other aspects of the program.

### ****How are two objects compared in PHP?****

PHP provides you with the ‘==’ operator, which is used to compare two objects at a time. This is used to check if there is a common presence of attributes and values between the objects in comparison.

The ‘===’ operator is also used to compare if both objects in consideration are referencing to the same class.

### ****What is the meaning of break and continue statements in PHP?****

**Break:** This statement is used in a looping construct to terminate the execution of the iteration and to immediately execute the next code snippet outside the scope of the iterating statement.

**Continue**: This statement is used to skip the current iteration of the loop and continue to execute the next iteration until the looping construct is exited.

### ****How does JavaScript interact with PHP?****

JavaScript is a client-side programming language, while PHP is a server-side scripting language. PHP has the ability to generate JavaScript variables, and this can be executed easily in the browser, thereby making it possible to pass variables to PHP using a simple URL.

### ****How does the ‘foreach’ loop work in PHP?****

The foreach statement is a looping construct used in PHP to iterate and loop through the array data type. The working of foreach is simple; with every single pass of the value, elements get assigned a value and pointers are incremented. This process is repeated until the end of the array.

The following is the syntax for using the foreach statement in PHP:

foreach(array)

{

Code inside the loop;

}

### ****Differentiate between require() and require\_once() functions.****

|  |  |
| --- | --- |
| **require()** | **require\_once()** |
| The inclusion and evaluation of files | Includes files if they are not included before |
| Preferred for files with fewer functions | Preferred when there are a lot of functions |

**Is it possible to set infinite execution time in PHP?**

Yes, it is possible to have an infinite execution time in PHP for a script by adding the **set\_time\_limit(0)** function to the beginning of a script.

This can also be executed in the  ‘***php.ini’***file if not at the beginning of the script.

**What is the most used method for hashing passwords in PHP?**

The **crypt()** function is widely used for this functionality as it provides a large amount of hashing algorithms that can be used. These algorithms include [**md5**](https://intellipaat.com/blog/what-is-md5-algorithm/), sha1 or sha256.

### ****Differentiate between an indexed array and an associative array.****

Indexed arrays have elements that contain a numerical index value.

Example: $color=array("red","green","blue");

Here, red is at index 0, green at 1, and blue at 2.

Associative arrays, on the other hand, hold elements with string indices as shown below:

Example: $salary=array("Jacob"=>"20000","John"=>"44000","Josh"=>"60000");

### ****What are sessions and cookies in PHP?****

Sessions are global variables that are stored on the server inside the architecture. Every single session is tagged with a unique server ID that is later used to work with the storage and retrieval of values.

Cookies are entities used to identify unique users in the architecture. It is a small file that the server plants into the client system. This is done to get useful information from the client for the development of various aspects of the server.

### ****Is typecasting supported in PHP?****

Yes, typecasting is supported by PHP and can be done very easily. Following are the types that can be cast in PHP:

* **(int), (integer):** Cast to integer
* **(bool), (boolean):** Cast to boolean
* **(float), (double), (real):** Cast to float
* **(string):** Cast to string
* **(array):** Cast to array
* **(object):** Cast to object

### ****Can a form be submitted in PHP without making use of a submit button?****

Yes, a form can be submitted without the explicit use of a button. This is done by making use of the JavaScript **submit()** function easily.

### ****Does PHP support variable length argument functions?****

Yes, **PHP** supports the use of variable-length argument functions. This simply means that you can pass any number of arguments to a function. The syntax simply involves using three dots before the argument name as shown in the following example:

<?php

function add(...$num) {

$sum = 0;

foreach ($num as $n) {

$sum += $n;

}

return $sum;

}

echo add(5, 6, 7, 8);

?>

Output: 26

### ****What is the use of session\_start() and session\_destroy() functions?****

In PHP, the **session\_start()** function is used to start a new session. However, it can also resume an existing session if it is stopped. In this case, the return will be the current session if resumed.

Syntax:

session\_start();

The session\_destroy() function is mostly used to destroy all of the session variables as shown below:

<?php

session\_start();

session\_destroy();

?>

**How can you open a file in PHP?**

PHP supports file operations by providing users with a plethora of file-related functions.

In the case of opening a file, the **fopen()** function is used. This function can open a file or a URL as per requirement. It takes two arguments: **$filenam**e and **$mode**.

Syntax:

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

### ****What is the use of $message and $$message in PHP?****

Both **$message** and **$$message** are variables in PHP and the difference lies in the name. While $message is a variable with a fixed name, $$message is a variable with a name that is actually stored in $message.

Consider the following example:

If $message consists of ‘var’, then $message is nothing but ‘$var’.

Next up on these core PHP interview questions, you have to know a very important difference in PHP.

### ****Differentiate between GET and POST methods in PHP.****

|  |  |
| --- | --- |
| **GET Method** | **POST Method** |
| The GET method can only send a maximum of 1024 characters simultaneously | There is no restriction on the data size |
| GET does not support sending binary data | POST supports binary data as well as ASCII |
| QUERY\_STRING env variable is used to access the data that is sent | The HTTP protocol and the header are used to push the data |
| The $\_GET associative array is used to access the sent information | The $\_POST associative array is used to access the sent information here |

**What is the use of lambda functions in PHP?**

Being an anonymous function, the lambda function is used to first store data into a variable and then to pass it as arguments for the usage in other methods or functions.

Consider the following example:

$input = array(2, 5, 10);

$output = array\_filter($input, function ($x) { return $x > 2; });

The lambda function definition here:

function ($x) { return $x > 2; });

This is used further to store data into a variable, and then you can use it when required without the requirement of defining it again.

### ****Differentiate between compile-time exception and runtime exception in PHP.****

As the name suggests, if there is an occurrence of any sort of exception while the script is being compiled, it is called a compile-time exception. The FileNotFoundException is a good example of a compile-time exception.

An exception that interrupts the script while running is called a runtime exception. The ArrayIndexOutOfBoundException is an example of a runtime exception.

### ****What is the meaning of type hinting in PHP?****

Type hinting is used in PHP when there is a requirement to explicitly define the data type of an argument when passing it through a function.

When this function is first called, PHP will run a quick check to analyze the presence of all the data types that are specified. If it is different, then the runtime will stop as an exception will be raised.

Next up you have to understand how to connect a URL with PHP.

### ****How is a URL connected to PHP?****

Any URL can be connected to PHP easily by making use of the library called cURL. This comes as a default library with the standard installation of PHP.

The term ‘cURL’ stands for client-side URL, allowing users to connect to a URL and pick up information from that page to display.

### ****How does string concatenation work in PHP?****

String concatenation is carried out easily in PHP by making use of the dot(.) operator. Consider the following example:

<?php $string1="Welcome"; $string2="to Intellipaat"; echo $string1 . " " . $string2; ?>

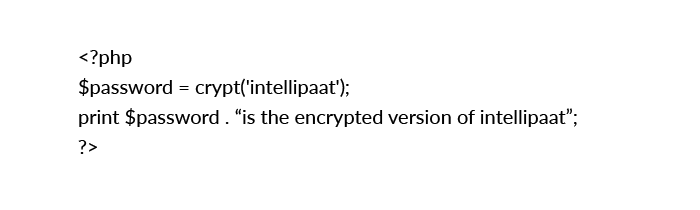
Output: Welcome to Intellipaat

### ****How can we encrypt a 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)

Here, $input\_string consists of the input string that has to be encrypted and $salt is also a string parameter that is used to generate a strong password and it acts as the base of the hashing process here. PHP uses DES for encryption. The format is as follows:



### ****Explain how to submit a form without a submit button.****

A form can be posted or submitted without the button in the following ways:

* On the OnClick event of a label in the form, a JavaScript function can be called to submit the form.  
  **Example:**

document.form\_name.submit()

* Using a Hyperlink: On clicking the link, a JavaScript function can be called.

Q5 php IQA code

* **A form can be submitted in the following ways as well without using the submit button:**
  + Submitting a form by clicking a link
  + Submitting a form by selecting an option from the drop-down box with the invocation of an onChange event
  + Using JavaScript:

document.form.submit();

* Using header(“location:page.php”);

**How can we increase the execution time of a PHP script?**

* The default time allowed for a PHP script to execute is 30 seconds as mentioned in the **php.ini** file. The function used is **set\_time\_limit(int sec)**. If the value passed is ‘0’, it takes unlimited time. It should be noted that if the default timer is set to 30 seconds and 20 seconds is specified in **set\_time\_limit()**, the script will run for 45 seconds.
* This time can be increased by modifying **max\_execution\_time** in seconds. The time must be changed keeping in mind the environment of the server. This is because modifying the execution time will affect all the sites hosted by the server.
* The script execution time can be increased by:
  + Using the **sleep()** function in the PHP script
  + Using the **set\_time\_limit()** function

The default limit is 30 seconds. The time limit can be set to zero to impose no time limit

### ****What is Zend Engine?****

Zend Engine is used internally by PHP as a compiler and runtime engine. PHP Scripts are loaded into memory and compiled into Zend OPCodes.

These OPCodes are executed and the HTML generated is sent to the client.

The Zend Engine provides memory and resource management and other standard services for the PHP language. Its performance, reliability, and extensibility have played a significant role in PHP’s increasing popularity.

### ****What library is used for PDF in PHP?****

The PDF functions in PHP can create PDF files using PDFlib version 6. PDFlib offers an object-oriented API for PHP5 in addition to the function-oriented API for PHP4.  
There is also the Â» Panda module.

FPDF is a PHP class, which allows generating PDF files with pure PHP (without using PDFlib). F from FPDF stands for Free: we may use it for any requirement and modify it to suit our needs. FPDF requires no extension (except zlib to activate compression and GD for GIF support) and works with PHP4 and PHP5.

### ****What are the new features introduced in PHP7?****

* Zend Engine 3 performance improvements and 64-bit integer support on Windows
* Uniform variable syntax
* AST-based compilation process
* Added Closure::call()
* Bitwise shift consistency across platforms
* (Null coalesce) operator
* Unicode codepoint escape syntax
* Return type declarations
* Scalar type (integer, float, string, and Boolean) declarations

### ****What is htaccess? Why do we use it and where?****

The .**htaccess** files are configuration files of Apache Server that provide a way to make configuration changes on a per-directory basis. A file, containing one or more configuration directives, is placed in a particular document directory; the directives apply to that directory and all subdirectories thereof.

These .htaccess files are used to change the functionality and features of the Apache web server.

**For instance:**

* The .htaccess file is used for URL rewrite.
* It is used to make the site password-protected.
* It can restrict some IP addresses so that on these restricted IP addresses, the site will not open.

### ****What is meant by PEAR in PHP?****

**PEAR** is an acronym for **PHP Extension and Application Repository**. The purpose of PEAR is to provide:

* A structured library of open-sourced code for PHP users
* A system for code distribution and package maintenance
* A standard style for writing code in PHP
* PHP Foundation Classes (PFC)
* PHP Extension Community Library (PECL)
* A website, mailing lists, and download mirrors to support the PHP/PEAR community

### ****Explain soundex() and metaphone().****

The **soundex()** function calculates the soundex key of a string. A soundex key is a 4-character long alphanumeric string that represents the English pronunciation of a word. The soundex() function can be used for spelling applications.

<?php

$str= “hello”;

Echo soundex($str);

?>

The **metaphone()** function calculates the metaphone key of a string. A metaphone key represents how a string sounds if it is pronounced by an English (native) person. This function can also be used for spelling applications.

<?php

echo metaphone(“world”);

?>

### ****What is Smarty?****

Smarty is a template engine written in PHP. Typically, these templates will include variables—like {$variable}—and a range of logical and loop operators to allow adaptability within the templates.

### ****What is Memcache?****

Memcache is a technology that caches objects in memory such that a web application can get to them really fast. It is used by sites, such as Digg, Facebook, and NowPublic, and is widely recognized as an essential ingredient in scaling any LAMP.

### ****How can we execute a PHP script using a command line?****

We just have to run the PHP CLI (Command-line Interface) program and provide the PHP script file name as the command-line argument, for example, **php myScript.php**, assuming **php** as the command to invoke the CLI program.

We have to keep in mind that if our PHP script is written for the Web CGI interface, it may not execute properly in the command-line environment.

**How can we know the number of days between two given dates using PHP?**

$date1 = date(‘Y-m-d’); $date2 = ‘2006-07-01’; $days = (strtotime($date1) – strtotime($date2)) / (60 \* 60 \* 24); echo “Number of days since ‘2006-07-01’: $days”;

**What is meant by urlencode and urldecode?**

urlencode() returns the URL encoded version of the given string. URL coding converts special characters into % signs followed by two hex digits.

For example: urlencode(“10.00%”) will return “10%2E00%25”. URL encoded strings are safe to be used as part of URLs.

urldecode() returns the URL decoded version of the given string.

**How to Get the Uploaded File Information in the Receiving Script?**

Uploaded file information is organized in $\_FILES as a two-dimensional array as:

$\_FILES[$fieldName][‘name’] – The Original file name on the browser system.

$\_FILES[$fieldName][‘type’] – The file type determined by the browser.

$\_FILES[$fieldName][‘size’] – The Number of bytes of the file content.

$\_FILES[$fieldName][‘tmp\_name’] – The temporary filename of the file in which the uploaded file was stored on the server.

$\_FILES[$fieldName][‘error’] – The error code associated with this file upload.

**What is the difference between mysql\_fetch\_object and mysql\_fetch\_array?**

MySQL fetch object will collect first single matching record where mysql\_fetch\_array will collect all matching records from the table in an array

**How do you pass a variable by value?**

Just like in C++, put an ampersand in front of it, like $a = &$b.

**What is the difference between ereg\_replace() and eregi\_replace()?**

eregi\_replace() function is identical to ereg\_replace() except that it ignores case distinction when matching alphabetic characters.

**How do I find out the number of parameters passed into function?**

func\_num\_args() function returns the number of parameters passed in.

**How can we extract string ‘abc.com ‘ from a string http://info@abc.com using regular** **expression of php?**

We can use the preg\_match() function with “/.\*@(.\*)$/” as the regular expression pattern. For example:

preg\_match(“/.\*@(.\*)$/”,”http://info@abc.com”,$data); echo $data[1];

**What is the difference between the functions unlink and unset?**

unlink() is a function for file system handling. It will simply delete the file in context. unset() is a function for variable management. It will make a variable undefined.

**How can we destroy the session, how can we unset the variable of a session?**

session\_unregister() – Unregister a global variable from the current session

session\_unset() – Free all session variables

**How can we know the count/number of elements of an array?**

a) sizeof($array) – This function is an alias of count()

b) count($urarray) – This function returns the number of elements in an array. Interestingly if you just pass a simple var instead of an array, count() will return 1

**How many values can the SET function of MySQL take?**

MySQL SET function can take zero or more values, but at the maximum it can take 64 values.

**What are the other commands to know the structure of a table using MySQL commands** **except EXPLAIN command?**

DESCRIBE table\_name;

**How can we find the number of rows in a table using MySQL?**

SELECT COUNT(\*) FROM table\_name;

**How can we find the number of rows in a result set using PHP?**

$result = mysql\_query($any\_valid\_sql, $database\_link);

$num\_rows = mysql\_num\_rows($result);

echo “$num\_rows rows found”;

**What is the difference between CHAR and VARCHAR data types?**

CHAR is a fixed length data type. CHAR(n) will take n characters of storage even if you enter less than n characters to that column. For example, “Hello!” will be stored as “Hello! ” in CHAR(10) column.

VARCHAR is a variable length data type. VARCHAR(n) will take only the required storage for the actual number of characters entered to that column. For example, “Hello!” will be stored as “Hello!” in VARCHAR(10) column.

**mysql\_fetch\_array(),  mysql\_fetch\_object(),  mysql\_fetch\_row()?**

**mysql\_fetch\_array**– Fetch a result row as an associative array and a numeric array.

**mysql\_fetch\_object** – Returns an object with properties that correspond to the fetched row and moves the internal data pointer ahead. Returns an object with properties that correspond to the fetched row, or FALSE if there are no more rows

**mysql\_fetch\_row()**– Fetches one row of data from the result associated with the specified result identifier. The row is returned as an array. Each result column is stored in an array offset, starting at offset 0.

**What is the difference between htmlentities() and htmlspecialchars()?**

htmlspecialchars() – Convert some special characters to HTML entities (Only the most widely used)

htmlentities() – Convert ALL special characters to HTML entities

**How can we get the properties (size, type, width, height) of an image using php image** **functions?**

image size use getimagesize() function

image width use imagesx() function

image height use imagesy() function

**Explain the ternary conditional operator in PHP?**

Expression preceding the ? is evaluated, if it’s true, then the expression preceding the : is executed, otherwise, the expression following : is executed.

**How many ways can we get the value of current session id?**

session\_id() returns the session id for the current session.

**How can we get the browser properties using php?**

<?php

echo $\_SERVER[‘HTTP\_USER\_AGENT’] . “\n\n”;

$browser = get\_browser(null, true);

print\_r($browser);

?>

**How can we know that a session is started or not?**

A session starts by session\_start()function. this session\_start() is always declared in header portion.it always declares first.then we write session\_register().

**What is the use of obj\_start()?**

Its intializing the object buffer, so that the whole page will be first parsed (instead of parsing in parts and thrown to browser gradually) and stored in output buffer so that after complete page is executed, it is thrown to the browser once at a time.

**39. What is the difference between Split and Explode?**

split()-used for JavaScript for processing the string and the explode()-used to convert

the String to Array, implode()-used for convert the array to String

Here the Example <?php $x=”PHP is a ServerSide Scripting Language”; $c=explode(” “,$x); print\_r($c); $d=implode(” “,$c); echo ” “.$d; ?> Javascript Example: list($month, $day, $year) = split(‘[/.-]’, $date);

**What is the use of sprintf() function?**

The sprintf() function writes a formatted string to a variable.

**What is the diffrence between Notify URL and Return URL?**

Notify URL is used to just notify the status while processing. Return URL is used to return after processing.

**What is the difference between ucfirst and ucwords?**

ucfirst() to convert the first letter of every string to uppercase, and ucwords(), to convert the first letter of every word in the string to uppercase.

**What is meant by nl2br()?**

nl2br() inserts a HTML tag <br> before all new line characters \n in a string.

**How To Read the Entire File into a Single String**?

<?php

$file = file\_get\_contents(“/windows/system32/drivers/etc/services”);

print(“Size of the file: “.strlen($file).”n”);

 ?>

**What are the different functions in sorting an array?**

Sorting functions in PHP: asort() arsort() ksort() krsort() uksort() sort() natsort() rsort()

**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

### 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

**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

### 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.

### 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.

### Explain PHP variable length argument function.

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

### 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.

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

### 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.

### 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.

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

**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()

### 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.

### 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 )

### How to download file in PHP?

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

 int readfile ( string $filename )

### 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.

### What is the use of the function ‘imagetypes()’?

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

### 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.

**What does the PHP error ‘Parse error in PHP – unexpected T\_variable at line x’ means?**

This is a PHP syntax error expressing that a mistake at the line x stops parsing and executing the program.

**What should we do to be able to export data into an Excel file?**

The most common and used way is to get data into a format supported by Excel. For example, it is possible to write a .csv file, to choose for example comma as a separator between fields and then to open the file with Excel.

**What is the function file\_get\_contents() useful for?**

file\_get\_contents() lets reading a file and storing it in a string variable.

**What is the function mysql\_pconnect() useful for?**

mysql\_pconnect() ensure a persistent connection to the database, it means that the connection does not close when the PHP script ends.

This function is not supported in PHP 7.0 and above

**How is it possible to know the number of rows returned in the result set?**

The function mysqli\_num\_rows() returns the number of rows in a result set.

**Which function gives us the number of affected entries by a query?**

mysqli\_affected\_rows() return the number of entries affected by an SQL query.

**How can we check the value of a given variable is a number?**

It is possible to use the dedicated function, is\_numeric() to check whether it is a number or not.

**How can we check the value of a given variable is alphanumeric?**

It is possible to use the dedicated function, ctype\_alnum to check whether it is an alphanumeric value or not.

**How do I check if a given variable is empty?**

If we want to check whether a variable has a value or not, it is possible to use the empty() function.

**How do I escape data before storing it in the database?**

The addslashes function enables us to escape data before storage into the database.

**How is it possible to remove escape characters from a string?**

The stripslashes function enables us to remove the escape characters before apostrophes in a string.

**How can we automatically escape incoming data?**

We have to enable the Magic quotes entry in the configuration file of PHP.

**What does the function get\_magic\_quotes\_gpc() means?**

The function get\_magic\_quotes\_gpc() tells us whether the magic quotes is switched on or no.

**Is it possible to remove the HTML tags from data?**

The strip\_tags() function enables us to clean a string from the HTML tags.

**How can we define a variable accessible in functions of a PHP script?**

This feature is possible using the global keyword.

**How is it possible to return a value from a function?**

A function returns a value using the instruction ‘return $value;’.

**Which cryptographic extension provide generation and verification of digital signatures?**

The PHP-OpenSSL extension provides several cryptographic operations including generation and verification of digital signatures.

**How is a constant defined in a PHP script?**

The define() directive lets us defining a constant as follows:

define ("ACONSTANT", 123);

**How is the ternary conditional operator used in PHP?**

It is composed of three expressions: a condition, and two operands describing what instruction should be performed when the specified condition is true or false as follows:

Expression\_1?Expression\_2 : Expression\_3;

**What is the function func\_num\_args() used for?**

The function func\_num\_args() is used to give the number of parameters passed into a function.

**In PHP, objects are they passed by value or by reference?**

In PHP, objects are passed by reference.

**Are Parent constructors called implicitly inside a class constructor?**

No, a parent constructor have to be called explicitly as follows:

parent::constructor($value)

**What’s the difference between \_\_sleep and \_\_wakeup?**

\_\_sleep returns the array of all the variables that need to be saved, while \_\_wakeup retrieves them.

**What is faster?**

1- Combining two variables as follows:

$variable1 = 'Hello ';

$variable2 = 'World';

$variable3 = $variable1.$variable2;

Or

2- $variable3 = "$variable1$variable2";

$variable3 will contain “Hello World”. The first code is faster than the second code especially for large large sets of data.

**When do sessions end?**

Sessions automatically end when the PHP script finishes executing but can be manually ended using the session\_write\_close().

**What is the difference between session\_unregister() and session\_unset()?**

The session\_unregister() function unregister a global variable from the current session and the session\_unset() function frees all session variables.

**What does $GLOBALS mean?**

$GLOBALS is associative array including references to all variables which are currently defined in the global scope of the script.

**What does $\_SERVER mean?**

$\_SERVER is an array including information created by the web server such as paths, headers, and script locations.

**What does $\_FILES means?**

$\_FILES is an associative array composed of items sent to the current script via the HTTP POST method..

**How can we get the error when there is a problem to upload a file?**

$\_FILES[‘userfile’][‘error’] contains the error code associated with the uploaded file.

**What does $\_ENV mean?**

$\_ENV is an associative array of variables sent to the current PHP script via the environment method.

**What does the scope of variables mean?**

The scope of a variable is the context within which it is defined. For the most part, all PHP variables only have a single scope. This single scope spans included and required files as well.

**what the difference between the ‘BITWISE AND’ operator and the ‘LOGICAL AND’ operator?**

$a and $b: TRUE if both $a and $b are TRUE.

$a & $b: Bits that are set in both $a and $b are set.

**What are the two main string operators?**

The first is the concatenation operator (‘.’), which returns the concatenation of its right and left arguments. The second is (‘.=’), which appends the argument on the right to the argument on the left.

**What does the array operator ‘===’ means?**

$a === $b TRUE if $a and $b have the same key/value pairs in the same order and of the same types.

**What is the differences between $a != $b and $a !== $b?**

!= means inequality (TRUE if $a is not equal to $b) and !== means non-identity (TRUE if $a is not identical to $b).

**How can we determine whether a PHP variable is an instantiated object of a certain class?**

To be able to verify whether a PHP variable is an instantiated object of a certain class we use instanceof.

**What is the goto statement useful for?**

The goto statement can be placed to enable jumping inside the PHP program. The target is pointed by a label followed by a colon, and the instruction is specified as a goto statement followed by the desired target label.

**what is the difference between Exception::getMessage and Exception:: getLine?**

Exception::getMessage lets us getting the Exception message and Exception::getLine lets us getting the line in which the exception occurred.

**What does the expression Exception::\_\_toString means?**

Exception::\_\_toString gives the String representation of the exception.

**How is it possible to parse a configuration file?**

The function parse\_ini\_file() enables us to load in the ini file specified in filename and returns the settings in it in an associative array.

**How can we determine whether a variable is set?**

The boolean function isset determines if a variable is set and is not NULL.

**What is the difference between the functions strstr() and stristr()?**

The string function strstr(string allString, string occ) returns part of allString from the first occurrence of occ to the end of allString. This function is case-sensitive. stristr() is identical to strstr() except that it is case insensitive.

**Is it possible to protect special characters in a query string?**

Yes, we use the urlencode() function to be able to protect special characters.

**What are the three classes of errors that can occur in PHP?**

The three basic classes of errors are notices (non-critical), warnings (serious errors) and fatal errors (critical errors).

**What is the default session time in PHP?**

The default session time in php is until the closing of the browser

**Is it possible to use COM component in PHP?**

Yes, it’s possible to integrate (Distributed) Component Object Model components ((D)COM) in PHP scripts which is provided as a framework.

**Explain how you can update Memcached when you make changes to PHP?**

When PHP changes you can update Memcached by

* **Clearing the Cache proactively:** Clearing the cache when an insert or update is made
* **Resetting the Cache:** It is similar to the first method but rather than just deleting the keys and waiting for the next request for the data to refresh the cache, reset the values after the insert or update.

### Differentiate between variables and constants in PHP

| **Variables** | **Constants** |
| --- | --- |
| The value of a variable can be changed during the execution. | The constant value can’t be changed during script execution. |
| Variables require compulsory usage of the $ sign at the start. | No dollar sign ($) is required before using a constant. |
| It is possible to define a variable by simple assignment. | Constants can’t be defined by simple assignments. They are defined using the define() function. |
| The default scope is the current access scope. | Constants can be accessed throughout without any scoping rules. |

### What is the difference between “echo” and “print” in PHP?

| **echo** | **print** |
| --- | --- |
| echo can output one or more strings. | print can only output one string and it always returns 1. |
| echo is faster than print because it does not return any value. | print is slower compared to echo. |
| If you want to pass more than one parameter to echo, a parenthesis should be used. | Use of parenthesis is not required with the argument list. |

### Tell me some of the disadvantages of PHP

* PHP is not suitable for giant content-based web applications.
* Since it is open-source, it is not secure. Because ASCII text files are easily available.
* Change or modification in the core behavior of online applications is not allowed by PHP.
* If we use more features of the PHP framework and tools, it will cause poor performance of online applications.
* PHP features a poor quality of handling errors. PHP lacks debugging tools, which are needed to look for warnings and errors. It has only a few debugging tools in comparison to other programming languages.

## Explain the importance of Parser in PHP?

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

You can parse PHP code with PHP using the token\_get\_all() function.

### How to connect to a URL in PHP?

Any URL can be connected to PHP easily by making use of the library called cURL. This comes as a default library with the standard installation of PHP.

The term cURL stands for client-side URL. cURL make use of libcurl(client-side URL Transfer Library) which supports many protocols like FTP, FTPS, HTTP/1, HTTP POST, HTTP PUT, HTTP proxy, HTTPS, IMAP, Kerberos etc. It allows you to connect to a URL and retrieve and display information from that page – like the HTML content of the page, HTTP headers, and their associated data, etc.

**Steps for connecting with URL using PHP cURL POST are given below:**

* Initialize cURL session.
* Define your URL where you want to post the request. We can directly enter this URL into the URL section inset option parameter or we can assign it to an object.
* Now, define the cURL options that you want to execute with the post option.
* After setting all the functions then it’s time to execute our cURL.
* After this, close the cURL and echo your object to check their response.

//Step 1 To initialize curl

$ch = curl\_init();

//Step 2 To set url where you want to post

$url = ‘http://www.localhost.com’;

//Step 3 Set curl functions which are needs to you

curl\_setopt($ch,CURLOPT\_URL,$url);

curl\_setopt($ch,CURLOPT\_POST,true);

curl\_setopt($ch,CURLOPT\_RETURNTRANSFER,true);

curl\_setopt($ch,CURLOPT\_POSTFIELD,’postv1 = value1&postv2 = value2’);

//Step 4 To execute the curl

$result = curl\_exec($ch);

//Step 5 Close curl

curl\_close($ch);

### How to create API in PHP?

API stands for Application Programming Interface. It defines the functions and variables. Communication between the database via PHP extensions is handled by API.

Now, REST API is the web architecture that uses HTTP protocol for exchanging data between two functions that means your application or system. Now, let us have a look at how to create REST API in PHP by considering the example of accessing data from a database using PHP script.

**Step 1 – Create a database:** To create a database run the query given below:

CREATE DATABASE phptest;

**Step 2 – Create a table:** After creating a database, you have to create a table with dummy data. To create a table run the query given below:

CREATE TABLE IF NOT EXISTS `transactions`

(

`id` int(20) NOT NULL AUTO\_INCREMENT,

`order\_id` int(50) NOT NULL,

`amount` decimal(9,2) NOT NULL,

`response\_code` int(10) NOT NULL,

`response\_desc` varchar(50) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `order\_id` (`order\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;

**Step 3 – Create a Database Connection:** Create a db.php file and paste the below-given database connection in it. Make sure to update these credentials with your database credentials.

<?php

// Enter your Host, username, password, database below.

$con = mysqli\_connect("localhost","root","","phptest");

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

die();

}

?>

**Step 4 – Create a REST API File:** Create an api.php file and copy the following script in it.

<?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,"Request is invalid");

}

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;

}

?>

The above code will accept the GET request and return output in the JSON format.

### Differentiate between GET and POST

| **GET** | **POST** |
| --- | --- |
| GET method is used for requesting data from a specified resource. | POST is used for sending the data to the server as a package in a separate communication with the processing script. |
| Data is sent in the form of URL parameters which are strings of name-value pairs separated by ampersands(&) | Data sent through the POST method will not be seen in the URL |
| GET method cannot be used for sending binary data like images or word documents | The POST method can be used to send ASCII as well as binary data like images and word documents |
| This method must not be used if you have any sensitive information like a password to be sent to the server. | Sensitive information can be sent using this method. |
| It can be used for submitting the form where the user can bookmark the result. | Submissions by form with POST cannot be bookmarked. |
| You can use this method only for data that is not secure. | Data sent through this method is secure. |
| GET method is not safer since parameters may be stored in web server logs or browser history. | POST method is safer than GET because the parameters are not stored in web server logs or browser history. |

# ****Why is object-oriented programming so popular?****

There are certain points due to which OOPs become so popular.

* It provides a better programming style, so you don’t need to write code again and again which you want to run, just make a class of the object and call it.
* As it supports the inheritance concept, an application created with OOPs can inherit another class property.
* It provides a modularity model that means if you change any part of code that is in a separate module, that won’t impact any other module.
* If you are stuck somewhere, this language allows your problems to break down into bite-sized pieces so that you can solve them.
* Due to its polymorphism feature, a single class can be used to create different objects, and that too from the same piece of code.

**What are the types of Class Variables?**

**We have three different types of Class Variables in OOPs.**

#### 1. Local Variables

These types of variables are declared locally in methods, blocks, and constructors. These variables are created when program control enters the methods, blocks, and constructor and are destroyed once program control leaves them.

#### 2. Instance Variables

These variables are declared outside a block, constructor, or method. These are created once a class object is created and destroyed when the object is destroyed.

#### 3. Static Variables

Static variables are also called class variables and are defined using the static keyword. These are declared within a class but outside a code block and a method. The creation of these variables starts when the program starts and is destroyed when the program ends.

**What is the difference between Method overriding and Method overloading?**

| **Method Overloading** | **Method Overriding** |
| --- | --- |
| It is the concept where we define two or more methods by the same name but with different signatures. | In method overriding we define two or more identical methods which have the same name and signatures. |
| The binding of the method is done at compile time. | The binding of the method is done at run time. |
| There are no class restrictions i.e. it can be achieved in the same or different classes. | There are class restrictions in it i.e. it can only be achieved in different classes. |

**What is virtual and pure virtual function?**

| **Virtual Function** | **Pure Virtual Function** |
| --- | --- |
| Virtual Function has its definition hidden in the Base class | They have no definition in the Base class. |
| The derived class can override the virtual function if required. | In the case of a pure virtual function, the derived class has to override it. |
| If the derived class fails to override then it has the option to use the virtual function of the base class | It will throw a compilation error if it fails to override pure virtual function. |

**List out some of the differences between a class and an object?**

| **CLASS** | **OBJECT** |
| --- | --- |
| It is a blueprint from which different instances (objects) are created. | Objects are the instances of the class. |
| Class is a logical entity. | Objects are physical entities. |
| Users can declare class only once. | Objects can be declared multiple times depending upon the requirements. |
| When a class is created there is no memory allocation | Objects allocated memory. |
| Class is a group of objects. | Objects are real-world entities such as pen, copy, mouse, etc. |
| Class can be declared using class keyword e.g class Student {} | Objects can be declared using the new keyword e.g. Student s1=new Student(); |

**What are the main features of OOPs?**

**You have to list these four features of OOPs while answering this question-**

* Inheritance
* Polymorphism
* Encapsulation
* Data Abstraction

**What is Encapsulation in oops with an example?**

The process of binding up data and functions together that manipulates them is known as encapsulation in OOPS. It protects the data from outside interference and misuse.

Let’s understand the concept of encapsulation with both real-world examples and with the help of a program.

It is an attribute of an object, and it contains all data which is hidden. That hidden data is restricted to the members of that class.

class Account {  
private int account\_number;  
private int account\_balance;

public void show Data() {  
//code to show data  
}

public void deposit(int a) {  
if (a < 0) {  
//show error  
} else  
account\_balance = account\_balance + a;  
}  
}

**What is Polymorphism and its types in OOPS?**

Polymorphism is one of the core concepts which is used in object-oriented programming. Polymorphism generally shows the relationships between parent class objects and child class objects.

### Types of polymorphism in OOPs

* **Compile Time Polymorphism**– It is also known as Static Binding and allows the programmer to implement various methods. Names can be the same but their parameters should be different.
* **Runtime Polymorphism**– It is also known as Dynamic Binding and allows the programmer to call a single overridden method during the runtime of the program.

**What is Inheritance?**

It is a technique in which one class acquires the property of another class. With the help of inheritance, we can reuse the fields and methods of the existing class.

**Inheritance** has three type, are given below.

* Single inheritance
* Multiple inheritance
* Multi level inheritance

But PHP supports only **single inheritance**, where only one class can be derived from single parent class. We can also use multiple inheritance by using **interfaces**.

**What is constructor and destructor?**

Constructor and Destructor both are special functions which are automatically called when an object is created and destroyed.

#### Example

**Constructor**

classAnimal

{

     public $name ="No-name animal";

     publicfunction \_\_construct(){

     echo "I'm alive!";

}

}

**Destructor**

classAnimal{

public $name ="No-name animal";

publicfunction \_\_construct($name){

echo "I'm alive!";

$this->name = $name;

}

publicfunction \_\_destruct(){

echo "I'm dead now :(";

}

}

$animal =newAnimal("Bob");

echo "Name of the animal: ". $animal->name;

**What is different types of Visibility?**

PHP have three access modifiers such as public, private and protected.

* public scope of this variable or function is available from anywhere, other classes and instances of the object.
* private scope of this variable or function is available in its own class only.
* protected scope of this variable or function is available in all classes that extend current class including the parent class.

**What are the Final class and Final methods?**

**Final Class**– A class that can’t be extended and inherited further is known as Final Class. This class is declared with the keyword final and should be declared.

**Final Method**– Methods in the final class are implicitly final and if a user uses the final keyword that means methods can’t be overridden by subclasses.

#### Example

class childClassname extends parentClassname {  
protected $numPages;

public function \_\_construct($author, $pages) {  
$this->\_author = $author;  
$this->numPages = $pages;  
}

final public function PageCount() {  
return $this->numPages;  
}  
}

**What is static keyword ?**

When a variable or function declared as static then it cannot be accessed with an instantiated class object. It is treats as public if no visibility is defined. It can also be used to define static variables and for late static bindings.

**What are the difference between overloading and overriding in oops?**

**Overloading**: It occurs when two or more methods in one class have the same method name but different parameters.

**Overriding :**It means having two methods with the same method name and parameters. One of the methods is in the parent class and the other is in the child class.

**What is “this”?**

It refers to the current object of a class.

**What are the difference between abstract class and interface in OOPS?**

Thera are many differences between abstract class and interface in php.

1. **Abstract methods** can declare with public, private and protected. But in case of **Interface methods** declared with public.

2. **Abstract classes** can have constants, members, method stubs and defined methods, but **interfaces**can only have constants and methods stubs.

3. **Abstract classes** doest not support multiple inheritance but **interface**support this.

4. **Abstract classes** can contain constructors but **interface**doest not support constructors.

**What is namespace in PHP?**

#### It allows us to use the same function or class name in different parts of the same program without causing a name collision.

namespace MyAPP;  
function output() {  
echo 'IOS!';  
}  
namespace MyNeWAPP;  
function output(){  
echo 'RSS!';  
}

**What is traits? How it is used in php?**

**Traits**is a group of methods that reuse in single inheritance. A Trait is intended to reduce some limitations of single inheritance by enabling a developer to reuse sets of methods.

#### Example

trait HelloWorld

{

use Hello, World;

}

class MyWorld {

use HelloWorld;

}

$world = new MyWorld();

echo $world–>sayHello() . ” “ . $world–>sayWorld(); //Hello World

**What are the advantages of OOPS in PHP?**

* Code Resusability
* Flexibility
* Maintainability
* Security
* Testability

**What is member function?**

Member function defined inside a class and are used to access object data.

**What is Encapsulation?**

Encapsulation is an attribute of an object, and it contains all data which is hidden.

That hidden data can be restricted to the members of that class. Levels are Public, Protected, Private, Internal and Protected Internal.

**What is Polymorphism?**

Polymorphism is nothing but assigning behaviour or value in a subclass to something that was already declared in the main class. Simply, polymorphism takes more than one form.

**What is Inheritance?**

Inheritance is a concept where one class shares the structure and behaviour defined in another class. If inheritance applied on one class is called Single Inheritance, and if it depends on multiple classes, then it is called multiple Inheritances.

**What are manipulators?**

Manipulators are the functions which can be used in conjunction with the insertion (<<) and extraction (>>) operators on an object. Examples are endl and setw.

**Define a constructor?**

Constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation. Rules for constructor are.

* · Constructor Name should be same as class name.
* · Constructor must have no return type.

**Define Destructor?**

Destructor is a method which is automatically called when the object is made of scope or destroyed. Destructor name is also same as class name but with the tilde symbol before the name.

**What is Inline function?**

Inline function is a technique used by the compilers and instructs to insert complete body of the function wherever that function is used in the program source code.

**What is friend function?**

Friend function is a friend of a class that is allowed to access to Public, private or protected data in that same class. If the function is defined outside the class cannot access such information.

Friend can be declared anywhere in the class declaration, and it cannot be affected by access control keywords like private, public or protected.

**What is function overloading?**

Function overloading is defined as a normal function, but it has the ability to perform different tasks. It allows creation of several methods with the same name which differ from each other by type of input and output of the function.

Example

void add(int& a, int& b);

void add(double& a, double& b);

void add(struct bob& a, struct bob& b);

**What is operator overloading?**

Operator overloading is a function where different operators are applied and depends on the arguments. Operator,–,\* can be used to pass through the function , and it has their own precedence to execute.

Example:

class complex { double real, imag;

public: complex(double r, double i) : real(r), imag(i) {}

complex operator+(complex a, complex b); complex operator\*(complex a, complex b); complex& operator=(complex a, complex b); }

a=1.2, b=6

**What is an abstract class?**

An abstract class is a class which cannot be instantiated. Creation of an object is not possible with abstract class, but it can be inherited. An abstract class can contain only Abstract method.

**What is a ternary operator?**

Ternary operator is said to be an operator which takes three arguments. Arguments and results are of different data types, and it is depends on the function. Ternary operator is also called as conditional operator.

**What is the use of finalize method?**

Finalize method helps to perform cleanup operations on the resources which are not currently used. Finalize method is protected, and it is accessible only through this class or by a derived class.

**What are different types of arguments?**

A parameter is a variable used during the declaration of the function or subroutine and arguments are passed to the function, and it should match with the parameter defined. There are two types of Arguments.

* · Call by Value – Value passed will get modified only inside the function, and it returns the same value whatever it is passed it into the function.
* · Call by Reference – Value passed will get modified in both inside and outside the functions and it returns the same or different value.

**What is super keyword?**

Super keyword is used to invoke overridden method which overrides one of its superclass methods. This keyword allows to access overridden methods and also to access hidden members of the superclass.

It also forwards a call from a constructor to a constructor in the superclass.

**What is method overriding?**

Method overriding is a feature that allows sub class to provide implementation of a method that is already defined in the main class. This will overrides the implementation in the superclass by providing the same method name, same parameter and same return type.

**What is an interface?**

An interface is a collection of abstract method. If the class implements an inheritance, and then thereby inherits all the abstract methods of an interface.

**What is exception handling?**

Exception is an event that occurs during the execution of a program. Exceptions can be of any type – Run time exception, Error exceptions. Those

exceptions are handled properly through exception handling mechanism like try, catch and throw keywords.

**What are tokens?**

Token is recognized by a compiler and it cannot be broken down into component elements. Keywords, identifiers, constants, string literals and operators are examples of tokens.

Even punctuation characters are also considered as tokens – Brackets, Commas, Braces and Parentheses.

**Difference between overloading and overriding?**

Overloading is static binding whereas Overriding is dynamic binding. Overloading is nothing but the same method with different arguments, and it may or may not return the same value in the same class itself.

Overriding is the same method names with same arguments and return types associates with the class and its child class.

**Difference between class and an object?**

An object is an instance of a class. Objects hold any information, but classes don’t have any information. Definition of properties and functions can be done at class and can be used by the object.

Class can have sub–classes, and an object doesn’t have sub–objects.

**What is an abstraction?**

Abstraction is a good feature of OOPS, and it shows only the necessary details to the client of an object. Means, it shows only necessary details for an object, not the inner details of an object. Example – When you want to switch on television, it not necessary to show all the functions of TV. Whatever is required to switch on TV will be showed by using abstract class.

**What are access modifiers?**

Access modifiers determine the scope of the method or variables that can be accessed from other various objects or classes. There are 5 types of access modifiers, and they are as follows.

* · Private.
* · Protected.
* · Public.
* · Friend.
* · Protected Friend.

**What is sealed modifiers?**

Sealed modifiers are the access modifiers where it cannot be inherited by the methods. Sealed modifiers can also be applied to properties, events and methods. This modifier cannot be applied to static members.

**How can we call the base method without creating an instance?**

Yes, it is possible to call the base method without creating an instance. And that method should be Static method.

Doing inheritance from that class. Use Base Keyword from derived class.

**What is the difference between new and override?**

The new modifier instructs the compiler to use the new implementation instead of the base class function. Whereas, Override modifier helps to override the base class function.

**What are the various types of constructors?**

There are three various types of constructors, and they are as follows:.

– Default Constructor – With no parameters.

– Parametric Constructor – With Parameters. Create a new instance of a

class and also passing arguments simultaneously.

– Copy Constructor – Which creates a new object as a copy of an existing object.

**What is early and late binding?**

Early binding refers to assignment of values to variables during design time whereas late binding refers to assignment of values to variables during run time.

**What is the difference between structure and a class?**

Structure default access type is public, but class access type is private. A structure is used for grouping data whereas class can be used for grouping data and methods. Structures are exclusively used for data and it doesn’t require strict validation, but classes are used to encapsulates and inherit data which requires strict validation.

**What are all the operators that cannot be overloaded?**

Following are the operators that cannot be overloaded –.

* 1. Scope Resolution (:: )
* 2. Member Selection (.)
* 3. Member selection through a pointer to function (.\*)

**What is dynamic or run time polymorphism?**

Dynamic or Run time polymorphism is also known as method overriding in which call to an overridden function is resolved during run time, not at the compile time. It means having two or more methods with the same name, same signature but with different implementation.

**What are base class, sub class and super class?**

Base class is the most generalized class, and it is said to be a root class.

Sub class is a class that inherits from one or more base classes.

Super class is the parent class from which another class inherits.

**What is static and dynamic binding?**

Binding is nothing but the association of a name with the class. Static binding is a binding in which name can be associated with the class during compilation time, and it is also called as early Binding. Dynamic binding is a binding in which name can be associated with the class during execution time, and it is also called as Late Binding.

**How to get current MySQL version?**

SELECT VERSION (); is used to get the current version of MySQL.

**What storage engines are used in MySQL?**

Storage engines are called table types and data is stored in files using various techniques.

Technique involves:

Storage mechanism Locking levels Indexing Capabilities and functions.

**What does a TIMESTAMP do on UPDATE CURRENT\_TIMESTAMP data type?**

TIMESTAMP column is updated with Zero when the table is created. UPDATE CURRENT\_TIMESTAMP modifier updates the timestamp field tocurrent time whenever there is a change in other fields of the table.

**What does myisamchk do?**

It compress the MyISAM tables, which reduces their disk or memory usage.

**How do you control the max size of a HEAP table?**

Maximum size of Heal table can be controlled by MySQL config variable called max\_heap\_table\_size.

**What is the difference between MyISAM Static and MyISAM Dynamic?**

In MyISAM static all the fields will have fixed width. The Dynamic MyISAM table will have fields like TEXT, BLOB, etc. to accommodate the data types with various lengths.

MyISAM Static would be easier to restore in case of corruption.

**What happens when the column is set to AUTO INCREMENT and if you reach maximum value in the table?**

It stops incrementing. Any further inserts are going to produce an error, since the key has been used already.

**How can we find out which auto increment was assigned on Last insert?**

LAST\_INSERT\_ID will return the last value assigned by Auto\_increment and it is not required to specify the table name.

**How can you see all indexes defined for a table?**

Indexes are defined for the table by:

SHOW INDEX FROM <tablename>;

**What do you mean by % and \_ in the LIKE statement?**

% corresponds to 0 or more characters, \_ is exactly one character in the LIKE statement.

**How can we convert between Unix & MySQL timestamps?**

UNIX\_TIMESTAMP is the command which converts from MySQL timestamp to Unix timestamp

FROM\_UNIXTIME is the command which converts from Unix timestamp to MySQL timestamp.

**How can we get the number of rows affected by query?**

Number of rows can be obtained by

[sql]SELECT COUNT (user\_id) FROM users;[/sql]

**What is the difference between the LIKE and REGEXP operators?**

LIKE and REGEXP operators are used to express with ^ and %.

[sql]SELECT \* FROM employee WHERE emp\_name REGEXP “^b”; SELECT \* FROM employee WHERE emp\_name LIKE “%b”;[/sql]

**What is the difference between mysql\_fetch\_array and mysql\_fetch\_object?**

Following are the differences between mysql\_fetch\_array and mysql\_fetch\_object:

mysql\_fetch\_array() -Returns a result row as an associated array or a regular array from database.

mysql\_fetch\_object – Returns a result row as object from database.

**How can we run batch mode in mysql?**

Following commands are used to run in batch mode:

[sql]mysql ; mysql mysql.out[/sql]

**Where MyISAM table will be stored and also give their formats of storage?**

Each MyISAM table is stored on disk in three formats:

**How MySQL Optimizes DISTINCT?**

DISTINCT is converted to a GROUP BY on all columns and it will be combined with ORDER BY clause.

[sql]SELECT DISTINCT t1.a FROM t1,t2 where t1.a=t2.a;[/sql]

**How to enter Characters as HEX Numbers?**

If you want to enter characters as HEX numbers, you can enter HEX numbers with single quotes and a prefix of (X), or just prefix HEX numbers with (Ox).

A HEX number string will be automatically converted into a character string, if the expression context is a string.

**How to display top 50 rows?**

In MySql, top 50 rows are displayed by using this following query:

[sql]SELECT \* FROM LIMIT 0,50[/sql]

**How many columns can be used for creating Index?**

Maximum of 16 indexed columns can be created for any standard table.

**What is the different between NOW() and CURRENT\_DATE()?**

NOW () command is used to show current year,month,date with hours,minutes and seconds.

CURRENT\_DATE() shows current year,month and date only.

**What are the objects can be created using CREATE statement?**

Following objects are created using CREATE statement:

DATABASE EVENT FUNCTION INDEX PROCEDURE TABLE TRIGGER USER VIEW

**How many TRIGGERS are allowed in MySql table?**

SIX triggers are allowed in MySql table. They are as follows:

BEFORE INSERT AFTER INSERT BEFORE UPDATE AFTER UPDATE BEFORE DELETE and AFTER DELETE

**What are the nonstandard string types?**

Following are Non-Standard string types:

TINYTEXT TEXT MEDIUMTEXT LONGTEXT

**What are all the Common SQL Function?**

CONCAT(A, B) – Concatenates two string values to create a single string output. Often used to combine two or more fields into one single field.

FORMAT(X, D) – Formats the number X to D significant digits.

CURRDATE(), CURRTIME() – Returns the current date or time.

NOW() – Returns the current date and time as one value.

MONTH(), DAY(), YEAR(), WEEK(), WEEKDAY() – Extracts the given data from a date value.

HOUR(), MINUTE(), SECOND() – Extracts the given data from a time value.

DATEDIFF(A, B) – Determines the difference between two dates and it is commonly used to calculate age

SUBTIMES(A, B) – Determines the difference between two times.

FROMDAYS(INT) – Converts an integer number of days into a date value.

**What does tee command do in MySQL?**

tee followed by a filename turns on MySQL logging to a specified file. It can be stopped by command notee.

**Can you save your connection settings to a conf file?**

Yes, and name it ~/.my.conf. You might want to change the permissions on the file to 600, so that it’s not readable by others.

**What are some good ideas regarding user security in MySQL?**

There is no user without a password. There is no user without a user name. There is no user whose Host column contains % (which here indicates that the user can log in from anywhere in the network or the Internet). There are as few users as possible (in the ideal case only root) who have unrestricted access.

**What is SERIAL data type in MySQL?**

BIGINT NOT NULL PRIMARY KEY AUTO\_INCREMENT

**What happens when the column is set to AUTO INCREMENT and you reach the maximum value for that table?**

It stops incrementing. It does not overflow to 0 to prevent data losses, but further inserts are going to produce an error, since the key has been used already.

**Explain the difference between BOOL, TINYINT and BIT.**

Prior to MySQL 5.0.3: those are all synonyms. After MySQL 5.0.3: BIT data type can store 8 bytes of data and should be used for binary data.

**Explain the difference between FLOAT, DOUBLE and REAL.**

FLOATs store floating point numbers with 8 place accuracy and take up 4 bytes. DOUBLEs store floating point numbers with 16 place accuracy and take up 8 bytes. REAL is a synonym of FLOAT for now.

**If you specify the data type as DECIMAL (5, 2), what’s the range of values that can go in this table?**

999.99 to -99.99. Note that with the negative number the minus sign is considered one of the digits.

**What happens if a table has one column defined as TIMESTAMP?**

That field gets the current timestamp whenever the row gets altered.

**But what if you really want to store the timestamp data, such as the publication date of the article?**

Create two columns of type TIMESTAMP and use the second one for your real data.

**Explain data type TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE** **CURRENT\_TIMESTAMP**

The column exhibits the same behavior as a single timestamp column in a table with no other timestamp columns.

**What does TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP data type do?**

On initialization places a zero in that column, on future updates puts the current value of the timestamp in.

**Explain TIMESTAMP DEFAULT ‘2006:09:02 17:38:44′ ON UPDATE** **CURRENT\_TIMESTAMP.**

 A default value is used on initialization, a current timestamp is inserted on update of the row.

**If I created a column with data type VARCHAR (3), what would I expect to see in MySQL table?**

CHAR(3), since MySQL automatically adjusted the data type.

**How would you write a query to select all teams that won either 2, 4, 6 or 8 games?**

SELECT team\_name FROM teams WHERE team\_won IN (2, 4, 6, 8). How would you select all the users, whose phone number is null? SELECT user\_name FROM users WHERE ISNULL(user\_phonenumber);

**What are ENUMs used for in MySQL?**

You can limit the possible values that go into the table. CREATE TABLE months (month ENUM ’January’, ’February’, ’March’,); INSERT months VALUES (’April’).13. What is the difference between CHAR\_LENGTH and

LENGTH? The first is, naturally, the character count. The second is byte count. For the Latin characters the numbers are the same, but they’re not the same for Unicode and other encodings.

**How quoting and escaping work in SELECT QUERY?**

SELECT ‘hello’, ‘“hello”’,‘““hello””’, ‘hel‘‘lo’, ‘\‘hello’.

**How we get Sum of column?**

SELECT SUM(\*) FROM [table name];

**How would you change a table to InnoDB?**

ALTER TABLE name\_file ENGINE innodb;

**How do you concatenate strings in MySQL?**

CONCAT (string1, string2, string3)

**How do you get the month from a timestamp?**

SELECT MONTH(january\_timestamp) from tablename;

**What do % and \_ mean inside LIKE statement?**

% corresponds to 0 or more characters, \_ is exactly one character.

**How do you get the current date in Mysql?**

SELECT CURRENT\_DATE();

**You wrote a search engine that should retrieve 10 results at a time, but at the same time you’d** **like to know how many rows there’re total. How do you display that to the user?**

SELECT SQL\_CALC\_FOUND\_ROWS page\_title FROM web\_pages LIMIT 1,10; SELECT FOUND\_ROWS();

**What does this query mean: SELECT user\_name, user\_isp FROM users LEFT JOIN isps USING** **(user\_id)?**

It’s equivalent to saying SELECT user\_name, user\_isp FROM users LEFT JOIN isps WHERE users.user\_id=isps.user\_id

**How do you display the list of database in mysql?**

SHOW DATABASES;

**How do you display the structure of the table?**

DESCRIBE table\_name;

**How do you find out which auto increment was assigned on the last insert?**

SELECT LAST\_INSERT\_ID() will return the last value assigned by the auto\_increment function. Note that you don’t have to specify the table name.

**What is Like operator for and what are wild cards?**

LIKE operator is used to match patterns. A “%” sign is used to define the pattern. Below SQL statement will return all words with letter “S”

SELECT \* FROM pcdsEmployee WHERE EmpName LIKE ‘S%’

Below SQL statement will return all words which end with letter “S”

SELECT \* FROM pcdsEmployee WHERE EmpName LIKE ‘%S’

Below SQL statement will return all words having letter “S” in between

SELECT \* FROM pcdsEmployee WHERE EmpName LIKE ‘%S%’

“\_” operator (we can read as “Underscore Operator”). “\_” operator is the character defined at that point. In the below sample fired a query

Select name from pcdsEmployee where name like ‘\_s%’

So all name where second letter is “s” is returned.

**What is the SQL ” IN ” clause?**

SQL IN operator is used to see if the value exists in a group of values. For instance the below SQL checks if the Name is either ‘rohit’ or ‘Anuradha’

SELECT \* FROM pcdsEmployee WHERE name IN (‘Rohit’,’Anuradha’)

Also you can specify a not clause with the same.

SELECT \* FROM pcdsEmployee WHERE age NOT IN (17,16)

**How to select the first record in a given set of rows?**

Select top 1 \* from sales.salesperson

**What is a self-join?**

If we want to join two instances of the same table we can use self-join.

### What are the technical specifications of MySQL?

MySQL has the following technical specifications –

* Flexible structure
* High performance
* Manageable and easy to use
* Replication and high availability
* Security and storage management
* Drivers
* Graphical Tools
* MySQL Enterprise Monitor
* MySQL Enterprise Security
* JSON Support
* Replication & High-Availability
* Manageability and Ease of Use
* OLTP and Transactions
* Geo-Spatial Support

### Why do we use the MySQL database server?

First of all, the MYSQL server is free to use for developers and small enterprises.

MySQL server is open source.

MySQL’s community is tremendous and supportive; hence any help regarding MySQL is resolved as soon as possible.

MySQL has very stable versions available, as MySQL has been in the market for a long time. All bugs arising in the previous builds have been continuously removed, and a very stable version is provided after every update.

The MySQL database server is very fast, reliable, and easy to use. You can easily use and modify the software. MySQL software can be downloaded free of cost from the internet.

### What are the different tables present in MySQL?

There are many tables that remain present by default. But, MyISAM is the default database engine used in MySQL. There are five types of tables that are present:

* MyISAM
* Heap
* Merge
* INNO DB
* ISAM

### How to install MySQL?

Installing MySQL on our system allows us to safely create, drop, and test web applications without affecting our live website’s data. There are many ways to use MySQL on our system, but the best way is to install it manually. The manual installation allows us to learn more about the system and provides more control over the database. To see the installation steps of MySQL in Windows goes to the below link:

<https://www.javatpoint.com/how-to-install-mysql>

Manual installation of MySQL has several benefits:

* Backing up, reinstalling, or moving databases from one location to another can be achieved in a second.
* It provides more control to how and when MySQL server starts and closes.
* We can install MySQL anywhere, like in a portable USB drive.

### How to add foreign keys in MySQL?

The foreign key is used to link one or more tables together. It matches the primary key field of another table to link the two tables. It allows us to create a parent-child relationship with the tables. We can add a foreign key to a table in two ways:

* Using the CREATE TABLE Statement
* Using the ALTER TABLE Statement

Following is the syntax to define a foreign key using CREATE TABLE OR ALTER TABLE statement:

1. [CONSTRAINT constraint\_name]
2. FOREIGN KEY [foreign\_key\_name] (col\_name, …)
3. REFERENCES parent\_tbl\_name (col\_name,…)

.**How to connect to the MySQL database?**

MySQL allows us to connect with the database server in mainly two ways:

**Using Command-line Tool**

We can find the command-line client tool in the bin directory of the [MySQL’s installation](https://www.javatpoint.com/how-to-install-mysql)

folder. To invoke this program, we need to navigate the installation folder’s bin directory and type the below command:

mysql

Next, we need to run the below command to connect to the MySQL Server:

 shell>mysql -u root -p

Finally, type the password for the selected user account root and press Enter:

 Enter password: \*\*\*\*\*\*\*\*

After successful connection, we can use the below command to use the:

 USE database\_name;

**Using MySQL Workbench**

We can make a connection with database using [MySQL Workbench](https://www.javatpoint.com/mysql-workbench)

, simply clicking the plus (+) icon or navigating to the menu bar -> Database -> Connect to Database, the following screen appears. Now, you need to fill all the details to make a connection:

Once we finished this setup, it will open the MySQL Workbench screen. Now, we can double click on the newly created connection to connect with the database server.

To read more information, [click here](https://www.javatpoint.com/mysql-workbench)

### How to change the MySQL password?

We can change the MySQL root password using the below statement in the new notepad file and save it with an appropriate name:

 ALTER USER ‘root’@‘localhost’ IDENTIFIED BY ‘NewPassword’;

Next, open a Command Prompt and navigate to the MySQL directory. Now, copy the following folder and paste it in our DOS command and press the Enter key.

 C:\Users\javatpoint> CD C:\Program Files\MySQL\MySQL Server 8.0\bin

Next, enter this statement to change the password:

 mysqld –init-file=C:\\mysql-notepadfile.txt

Finally, we can log into the MySQL server as root using this new password. After launches the MySQL server, it is to delete the C:\myswl-init.txt file to ensure the password change.

### How to create a database in MySQL Workbench?

To create a new database in MySQL Workbench, we first need to launch the MySQL Workbench and log in using the username and password. Go to the Navigation tab and click on the Schema menu. Right-click under the Schema menu and select Create Schema or click the database icon (red rectangle), as shown in the following screen.

A new popup screen appears where we need to fill all the details. After entering the details, click on the Apply button and then the Finish button to complete the database creation.

### How to create a table in MySQL Workbench?

Launch the MySQL Workbench and go to the Navigation tab and click on the Schema menu where all the previously created databases are shown. Select any database and double click on it. It will show the sub-menus where we need to select the Tables option.

Select Tables sub-menu, right-click on it and select Create Table option. We can also click on create a new table icon (shown in red rectangle) to create a table. It will open the new popup screen where we need to fill all the details to create a table. Here, we will enter the table name and column details. After entering the details, click on the Apply button and then the Finish button to complete the table creation.

### How to change the table name in MySQL?

Sometimes our table name is non-meaningful. In that case, we need to change or rename the table name. MySQL provides the following syntax to rename one or more tables in the current database:

 mysql> RENAME old\_table TO new\_table;

If we want to change more than one table name, use the below syntax:

 RENAME TABLE old\_tab1 TO new\_tab1,

old\_tab2 TO new\_tab2, old\_tab3 TO new\_tab3;

### How to change the database name in MySQL?

Sometimes we need to change or rename the database name because of its non-meaningful name. To rename the database name, we need first to create a new database into the MySQL server. Next, MySQL provides the mysqldump shell command to create a dumped copy of the selected database and then import all the data into the newly created database. The following is the syntax of using mysqldump command:

 mysqldump -u username -p “password” -R oldDbName > oldDbName.sql

Now, use the below command to import the data into the newly created database:

 mysql -u username -p“password” newDbName < oldDbName.sql

### How to import a database in MySQL?

Importing database in MySQL is a process of moving data from one place to another place. It is a very useful method for backing up essential data or transferring our data between different locations. For example, we have a contact book database, which is essential to keep it in a secure place. So we need to export it in a safe place, and whenever it lost from the original location, we can restore it using import options.

In MySQL, we can import a database in mainly two ways:

* Command Line Tool
* MySQL Workbench

### How to change the column name in MySQL?

While creating a table, we have kept one of the column names incorrectly. To change or rename an existing column name in MySQL, we need to use the ALTER TABLE and CHANGE commands together. The following are the syntax used to rename a column in MySQL:

 ALTER TABLE table\_name

CHANGE COLUMN old\_column\_name new\_column\_name column\_definition [FIRST|AFTER existing\_column];

Suppose the column’s current name is S\_ID, but we want to change this with a more appropriate title as Stud\_ID. We will use the below statement to change its name:

 ALTER TABLE Student CHANGE COLUMN S\_ID Stud\_ID varchar(10);

### How to delete columns in MySQL?

We can remove, drop, or delete one or more columns in an existing table using the ALTER TABLE statement as follows:

 ALTER TABLE table\_name DROP COLUMN column\_name1, column\_name2….;

### How to insert data in MySQL?

We can insert data in a MySQL table using the INSERT STATEMENT. This statement allows us to insert single or multiple rows into a table. The following is the basic syntax to insert a record into a table:

 INSERT INTO table\_name ( field1, field2,…fieldN )

VALUES  ( value1, value2,…valueN );

If we want to insert more than one rows into a table, use the below syntax:

 INSERT INTO table(field1, field2,…fieldN)

1. VALUES
2. (value1, value 2, …),
3. (value1, value2, …),
4. …
5. (value1, value2, …);

.

### How to delete a row in MySQL?

We can delete a row from the MySQL table using the DELETE STATEMENT within the database. The following is the generic syntax of DELETE statement in MySQL to remove one or more rows from a table:

 DELETE FROM table\_name WHERE Condition\_specified;

It is noted that if we have not specified the WHERE clause with the syntax, this statement will remove all the records from the given table.

### How to join two tables in MySQL?

We can connect two or more tables in MySQL using the JOIN clause. MySQL allows various types of JOIN clauses. These clauses connect multiple tables and return only those records that match the same value and property in all tables. The following are the four easy ways to join two or more tables in MySQL:

* Inner Join
* Left Join
* Right Join
* Cross Join

### How to join three tables in MySQL?

Sometimes we need to fetch data from three or more tables. There are two types available to do these types of joins. Suppose we have three tables named Student, Marks, and Details.

Let’s say Student has (stud\_id, name) columns, Marks has (school\_id, stud\_id, scores) columns, and Details has (school\_id, address, email) columns.

**1. Using SQL Join Clause**

This approach is similar to the way we join two tables. The following query returns result from three tables:

1. SELECT name, scores, address, email FROM Student s
2. INNER JOIN Marks m on s.stud\_id = m.stud\_id
3. INNER JOIN Details d on d.school\_id = m.school\_id;

**2. Using Parent-Child Relationship**

It is another approach to join more than two tables. In the above tables, we have to create a parent-child relationship. First, create column X as a primary key in one table and as a foreign key in another table. Therefore, stud\_id is the primary key in the Student table and will be a foreign key in the Marks table. Next, school\_id is the primary key in the Marks table and will be a foreign key in the Details table. The following query returns result from three tables:

1. SELECT name, scores, address, email
2. FROM Student s, Marks m, Details d
3. WHERE s.stud\_id = m.stud\_id AND m.school\_id = d.school\_id;

### How to update the table in MySQL?

We can update existing records in a table using the UPDATE statement that comes with the SET and WHERE clauses. The SET clause changes the values of the specified column. The WHERE clause is optional, which is used to specify the condition. This statement can also use to change values in one or more columns of a single row or multiple rows at a time. Following is a generic syntax of UPDATE command to modify data into the MySQL table:

1. UPDATE table\_name
2. SET field1=new-value1, field2=new-value2, …
3. [WHERE Clause]

### What is MySQL Workbench?

MySQL Workbench is a unified visual database designing or GUI tool used for working on MySQL databases. It is developed and maintained by Oracle that provides SQL development, data migration, and comprehensive administration tools for server configuration, user administration, backup, etc. We can use this Server Administration to create new physical data models, E-R diagrams, and SQL development. It is available for all major operating systems. MySQL provides supports for it from MySQL Server version v5.6 and higher.

It is mainly available in three editions, which are given below:

* Community Edition (Open Source, GPL)
* Standard Edition (Commercial)
* Enterprise Edition (Commercial)

### How to drop the primary key in MySQL?

MySQL primary key is a single or combination of the field used to identify each record in a table uniquely. A primary key column cannot be null or empty. We can remove or delete a primary key from the table using the ALTER TABLE statement. The following syntax is used to drop the primary key:

 ALTER TABLE table\_name  DROP PRIMARY KEY;

### How to create a Stored Procedure in MySQL?

A stored procedure is a group of SQL statements that we save in the database. The SQL queries, including INSERT, UPDATE, DELETE, etc. can be a part of the stored procedure. A procedure allows us to use the same code over and over again by executing a single statement. It stores in the database data dictionary.

We can create a stored procedure using the below syntax:

1. CREATE PROCEDURE procedure\_name [ (parameter datatype [, parameter datatype]) ]
2. BEGIN
3. Body\_section of SQL statements
4. END;

This statement can return one or more value through parameters or may not return any result. The following example explains it more clearly:

1. DELIMITER $$
2. CREATE PROCEDURE get\_student\_info()
3. BEGIN
4. SELECT \* FROM Student\_table;
5. END$$

### How to execute a stored procedure in MySQL?

We can execute a stored procedure in MySQL by simply CALL query. This query takes the name of the stored procedure and any parameters we need to pass to it. The following is the basic syntax to execute a stored procedure:

1. CALL stored\_procedure\_name (argument\_list);

Let’s understand it with this example:

1. CALL Product\_Pricing (@pricelow, @pricehigh);

Here, a stored procedure named Product\_Pricing calculates and returns the lowest and highest product prices.

### How to create a View in MySQL?

A view is a database object whose values are based on the base table. It is a **virtual table** created by a query by joining one or more tables. It is operated similarly to the base table but does not contain any data of its own. If any changes occur in the underlying table, the same changes reflected in the View also.

Following is the general syntax of creating a VIEW in MySQL:

1. CREATE [OR REPLACE] VIEW view\_name AS
2. SELECT columns
3. FROM tables
4. [WHERE conditions];

### How to create a Trigger in MySQL?

A trigger is a procedural code in a database that automatically invokes whenever certain events on a particular table or view in the database occur. It can be executed when records are inserted into a table, or any columns are being updated. We can create a trigger in MySQL using the syntax as follows:

1. CREATE TRIGGER trigger\_name
2. [before | after]
3. {insert | update | delete}
4. ON table\_name [FOR EACH ROW]
5. BEGIN
6. –variable declarations
7. –trigger code
8. END;

### How to clear screen in MySQL?

If we use MySQL in Windows, it is not possible to clear the screen before version 8. At that time, the Windows operating system provides the only way to clear the screen by exiting the MySQL command-line tool and then again open MySQL.

After the release of MySQL version 8, we can use the below command to clear the command line screen:

 mysql> SYSTEM CLS;

### How to create a new user in MySQL?

A USER in MySQL is a record in the USER-TABLE. It contains the login information, account privileges, and the host information for MySQL account to access and manage the databases. We can create a new user account in the database server using the MySQL Create User statement. It provides authentication, SSL/TLS, resource-limit, role, and password management properties for the new accounts.

The following is the basic syntax to create a new user in MySQL:

 CREATE USER [IF NOT EXISTS] account\_name IDENTIFIED BY ‘password’;

### How to check USERS in MySQL?

If we want to manage a database in MySQL, it is required to see the list of all user’s accounts in a database server. The following command is used to check the list of all users available in the database server:

 mysql> SELECT USER FROM mysql.user;

### How to import a CSV file in MySQL?

MySQL allows us to import the CSV (comma separated values) file into a database or table. A CSV is a plain text file that contains the list of data and can be saved in a tabular format. MySQL provides the LOAD DATA INFILE statement to import a CSV file. This statement is used to read a text file and import it into a database table very quickly. The full syntax to import a CSV file is given below:

1. LOAD DATA INFILE ‘C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/filename.csv’
2. INTO TABLE tablename
3. FIELDS TERMINATED BY ‘,’
4. OPTIONALLY ENCLOSED BY ‘”‘
5. LINES TERMINATED BY ‘\r\n’
6. IGNORE 1 ROWS;

### How to insert Date in MySQL?

MySQL allows us to use the INSERT STATEMENT to add the date in MySQL table. MySQL provides several data types for storing dates such as DATE, TIMESTAMP, DATETIME, and YEAR. The default format of the date in MySQL is YYYY-MM-DD. Following is the basic syntax to insert date in MySQL table:

 INSERT INTO table\_name (column\_name, column\_date) VALUES (‘DATE: Manual Date’, ‘2008-7-04’);

If we want to insert a date in the mm/dd/yyyy format, it is required to use the below statement:

1. INSERT INTO table\_name VALUES (STR\_TO\_DATE(date\_value, format\_specifier));

### How to check database size in MySQL?

MySQL allows us to query the information\_schema.tables table to get the information about the tables and databases. It will return the information about the data length, index length, collation, creation time, etc. We can check the size of the database on the server using the below syntax:

1. SELECT table\_schema AS ‘Database Name’,
2. SUM(data\_length + index\_length) ‘Size in Bytes’,
3. ROUND(SUM(data\_length + index\_length) / 1024 / 1024, 2) ‘Size in MB’
4. FROM information\_schema.tables
5. WHERE table\_schema = ‘testdb’
6. GROUP BY table\_schema;

If we want to check the size of the table in a specific database, use the following statement:

1. SELECT table\_name AS ‘Table Name’,
2. ROUND(((data\_length + index\_length) / 1024 / 1024), 2) AS ‘Size in MB’
3. FROM information\_schema.TABLES
4. WHERE table\_schema = ‘testdb’
5. ORDER BY (data\_length + index\_length) DESC;

### How does indexing works in MySQL?

Indexing is a process to find an unordered list into an ordered list. It helps in maximizing the query’s efficiency while searching on tables in MySQL. The working of MySQL indexing is similar to the book index.

Suppose we have a book and want to get information about, say, searching. Without indexing, it is required to go through all pages one by one, until the specific topic was not found. On the other hand, an index contains a list of keywords to find the topic mentioned on pages. Then, we can flip to those pages directly without going through all pages.

### Who owns MySQL?

MySQL is the most popular free and open-source database software which comes under the GNU General Public License. In the beginning, it was owned and sponsored by the Swedish company MySQL AB. Now, it is bought by Sun Microsystems (now Oracle Corporation), who is responsible for managing and developing the database.

### How to view the database in MySQL?

Working with the MySQL server, it is a common task to view or list the available databases. We can view all the databases on the MySQL server host using the following command:

 mysql> SHOW DATABASES;

### How to set auto increment in MySQL?

Auto Increment is a constraint that automatically generates a unique number while inserting a new record into the table. Generally, it is used for the primary key field in a table. In MySQL, we can set the value for an AUTO\_INCREMENT column using the ALTER TABLE statement as follows:

 ALTER TABLE table\_name AUTO\_INCREMENT = value;

### How to find the second highest salary in MySQL?

MySQL uses the LIMIT keyword, which can be used to limit the result set. It will allow us to get the first few rows, last few rows, or range of rows. It can also be used to find the second, third, or nth highest salary. It ensures that you have use order by clause to sort the result set first and then print the output that provides accurate results. The following query is used to get the second highest salary in MySQL:

1. SELECT salary
2. FROM (SELECT salary FROM employees ORDER BY salary DESC LIMIT 2) AS Emp ORDER BY salary LIMIT 1;

There are some other ways to find the second highest salary in MySQL, which are given below:

This statement uses subquery and IN clause to get the second highest salary:

1. SELECT MAX(salary)
2. FROM employees
3. WHERE salary NOT IN ( SELECT Max(salary) FROM employees);

This query uses subquery and < operator to return the second highest salary:

1. SELECT MAX(salary) From employees
2. WHERE salary < ( SELECT Max(salary) FROM employees);

### What is the difference between TRUNCATE and DELETE in MySQL?

* TRUNCATE is a DDL command, and DELETE is a DML command.
* It is not possible to use Where command with TRUNCATE QLbut you can use it with DELETE command.
* TRUNCATE cannot be used with indexed views, whereas DELETE can be used with indexed views.
* The DELETE command is used to delete data from a table. It only deletes the rows of data from the table while truncate is a very dangerous command and should be used carefully because it deletes every row permanently from a table.

### How many Triggers are possible in MySQL?

There are only six Triggers allowed to use in the MySQL database.

1. Before Insert
2. After Insert
3. Before Update
4. After Update
5. Before Delete
6. After Delete

### What is the heap table?

Tables that are present in memory is known as HEAP tables. When you create a heap table in MySQL, you should need to specify the TYPE as HEAP. These tables are commonly known as memory tables. They are used for high-speed storage on a temporary basis. They do not allow BLOB or TEXT fields.

### What is BLOB and TEXT in MySQL?

BLOB is an acronym that stands for a large binary object. It is used to hold a variable amount of data.

There are four types of the BLOB.

1. TINYBLOB
2. BLOB
3. MEDIUMBLOB
4. LONGBLOB

The differences among all these are the maximum length of values they can hold.

TEXT is a case-insensitive BLOB. TEXT values are non-binary strings (character string). They have a character set, and values are stored and compared based on the collation of the character set.

There are four types of TEXT.

1. TINYTEXT
2. TEXT
3. MEDIUMTEXT
4. LONGTEXT

### What is the difference between the heap table and the temporary table?

**Heap tables:**

Heap tables are found in memory that is used for high-speed storage temporarily. They do not allow BLOB or TEXT fields.

Heap tables do not support AUTO\_INCREMENT.

Indexes should be NOT NULL.

**Temporary tables:**

The temporary tables are used to keep the transient data. Sometimes it is beneficial in cases to hold temporary data. The temporary table is deleted after the current client session terminates.

**Main differences:**

The heap tables are shared among clients, while temporary tables are not shared.

Heap tables are just another storage engine, while for temporary tables, you need a special privilege (create temporary table).

### What is the difference between FLOAT and DOUBLE?

FLOAT stores floating-point numbers with accuracy up to 8 places and allocate 4 bytes. On the other hand, DOUBLE stores floating-point numbers with accuracy up to 18 places and allocates 8 bytes.

### What are the advantages of MySQL in comparison to Oracle?

1. MySQL is a free, fast, reliable, open-source relational database while Oracle is expensive, although they have provided Oracle free edition to attract MySQL users.
2. MySQL uses only just under 1 MB of RAM on your laptop, while Oracle 9i installation uses 128 MB.
3. MySQL is great for database enabled websites while Oracle is made for enterprises.
4. MySQL is portable.

### What are the disadvantages of MySQL?

1. MySQL is not so efficient for large scale databases.
2. It does not support COMMIT and STORED PROCEDURES functions version less than 5.0.
3. Transactions are not handled very efficiently.
4. The functionality of MySQL is highly dependent on other addons.
5. Development is not community-driven.

### What is the difference between CHAR and VARCHAR?

1. CHAR and VARCHAR have differed in storage and retrieval.
2. CHAR column length is fixed, while VARCHAR length is variable.
3. The maximum no. of character CHAR data types can hold is 255 characters, while VARCHAR can hold up to 4000 characters.
4. CHAR is 50% faster than VARCHAR.
5. CHAR uses static memory allocation, while VARCHAR uses dynamic memory allocation.

### What is the difference between MySQL\_connect and MySQL\_pconnect?

**Mysql\_connect:**

1. It opens a new connection to the database.
2. Every time you need to open and close the database connection, depending on the request.
3. Opens page whenever it is loaded.

**Mysql\_pconnect:**

1. In Mysql\_pconnect, “p” stands for persistent connection, so it opens the persistent connection.
2. The database connection cannot be closed.
3. It is more useful if your site has more traffic because there is no need to open and close connection frequently and whenever the page is loaded.

**What does “i\_am\_a\_dummy flag” do in MySQL?**

The “i\_am\_a\_dummy flag” enables the MySQL engine to refuse any UPDATE or DELETE statement to execute if the WHERE clause is not present. Hence it can save the programmer from deleting the entire table my mistake if he does not use WHERE clause.

**How to get the current date in MySQL?**

To get current date, use the following syntax:

 SELECT CURRENT\_DATE();

### What are the security alerts while using MySQL?

Install antivirus and configure the operating system’s firewall.

Never use the MySQL Server as the UNIX root user.

Change the root username and password Restrict or disable remote access.

**How to change a password for an existing user via mysqladmin?**

Mysqladmin -u root -p password “newpassword”.

### What is the difference between UNIX timestamps and MySQL timestamps?

Actually, both Unix timestamp and MySQL timestamp are stored as 32-bit integers, but MySQL timestamp is represented in the readable format of YYYY-MM-DD HH:MM:SS format.

### How to display the nth highest salary from a table in a MySQL query?

Let us take a table named the employee.

**To find Nth highest salary is:**

select distinct(salary)from employee order by salary desc limit n-1,1

**if you want to find 3rd largest salary:**

select distinct(salary)from employee order by salary desc limit 2,1

### What is the MySQL default port number?

MySQL default port number is 3306.

### What is REGEXP?

REGEXP is a pattern match using a regular expression. The regular expression is a powerful way of specifying a pattern for a sophisticated search.

Basically, it is a special text string for describing a search pattern. To understand it better, you can think of a situation of daily life when you search for .txt files to list all text files in the file manager. The regex equivalent for .txt will be .\*\.txt.

### How many columns can you create for an index?

You can a create maximum of 16 indexed columns for a standard table.

### What is the difference between NOW() and CURRENT\_DATE()?

NOW() command is used to show current year, month, date with hours, minutes, and seconds while CURRENT\_DATE() shows the current year with month and date only.

### Write a query to display the current date and time?

If you want to display the current date and time, use –

SELECT NOW();

If you want to display the current date only, use:

SELECT CURRENT\_DATE();

### What is the save point in MySQL?

A defined point in any transaction is known as savepoint.

SAVEPOINT is a statement in MySQL, which is used to set a named transaction savepoint with the name of the identifier.

### What is SQLyog?

SQLyog program is the most popular GUI tool for admin. It is the most popular MySQL manager and admin tool. It combines the features of MySQL administrator, phpMyadmin, and others. MySQL front ends and MySQL GUI tools.

### How do you backup a database in MySQL?

It is easy to back up data with phpMyAdmin. Select the database you want to backup by clicking the database name in the left-hand navigation bar. Then click the export button and make sure that all tables are highlighted that you want to back up. Then specify the option you want under export and save the output.

### Write a query to count the number of rows of a table in MySQL.

**SELECT COUNT** user\_id FROM users;

### Write a query to select all teams that won either 1, 3, 5, or 7 games.

**SELECT** team\_name FROM team WHERE team\_won IN (1, 3, 5, 7);

### How is the MyISAM table stored?

MyISAM table is stored on disk in three formats.

* ‘.frm’ file : storing the table definition
* ‘.MYD’ (MYData): data file
* ‘.MYI’ (MYIndex): index file

### What is the usage of ENUMs in MySQL?

ENUMs are string objects. By defining ENUMs, we allow the end-user to give correct input as in case the user provides an input that is not part of the ENUM defined data, then the query won’t execute, and an error message will be displayed which says “The wrong Query”. For instance, suppose we want to take the gender of the user as an input, so we specify ENUM(‘male’, ‘female’, ‘other’), and hence whenever the user tries to input any string any other than these three it results in an error.

ENUMs are used to limit the possible values that go in the table:

**For example:**

CREATE TABLE months (month ENUM ‘January’, ‘February’, ‘March’); INSERT months VALUES (‘April’).

### What are the advantages of MyISAM over InnoDB?

MyISAM follows a conservative approach to disk space management and stores each MyISAM table in a separate file, which can be further compressed if required. On the other hand, InnoDB stores the tables in the tablespace. Its further optimization is difficult.

### What are the differences between MySQL\_fetch\_array(), MySQL\_fetch\_object(), MySQL\_fetch\_row()?

Mysql\_fetch\_object is used to retrieve the result from the database as objects, while mysql\_fetch\_array returns result as an array. This will allow access to the data by the field names.

Using mysql\_fetch\_object field can be accessed as $result->name.

Using mysql\_fetch\_array field can be accessed as $result->[name].

Using mysql\_fetch\_row($result) where $result is the result resource returned from a successful query executed using the mysql\_query() function.

1. $result = mysql\_query(“SELECT \* from students”);
2. while($row = mysql\_fetch\_row($result))
3. {
4. Some statement;
5. }

### What is the difference between mysql\_connect and mysql\_pconnect?

Mysql\_connect() is used to open a new connection to the database, while mysql\_pconnect() is used to open a persistent connection to the database. It specifies that each time the page is loaded, mysql\_pconnect() does not open the database.

### What is the use of mysql\_close()?

Mysql\_close() cannot be used to close the persistent connection. However, it can be used to close a connection opened by mysql\_connect().

### What is MySQL data directory?

MySQL data directory is a place where MySQL stores its data. Each subdirectory under this data dictionary represents a MySQL database. By default, the information managed my MySQL = server mysqld is stored in the data directory.

### How do you determine the location of MySQL data directory?

The default location of MySQL data directory in windows is C:\mysql\data or C:\Program Files\MySQL\MySQL Server 5.0 \data.

### What is the usage of regular expressions in MySQL?

In MySQL, regular expressions are used in queries for searching a pattern in a string.

* \* Matches 0 more instances of the string preceding it.
* + matches one more instances of the string preceding it.
* ? Matches 0 or 1 instances of the string preceding it.
* . Matches a single character.
* [abc] matches a or b or z
* | separates strings
* ^ anchors the match from the start.
* “.” Can be used to match any single character. “|” can be used to match either of the two strings
* REGEXP can be used to match the input characters with the database.

**Example:**

The following statement retrieves all rows where column employee\_name contains the text 1000 (example salary):

1. Select employee\_name
2. From employee
3. Where employee\_name REGEXP ‘1000’
4. Order by employee\_name

**Explain Access Control Lists.**

An ACL is a list of permissions that are associated with an object. MySQL keeps the Access Control Lists cached in memory, and whenever the user tries to authenticate or execute a command, MySQL checks the permission required for the object, and if the permissions are available, then execution completes successfully.

### What is InnoDB ISAM?

InnoDB is a storage database for SQL. The ACID-transactions are also provided in InnoDB and also includes support for the foreign key. Initially owned by InnobaseOY now belongs to Oracle Corporation after it acquired the latter since 2005.

It is a system for file management developed by IBM, which allows records to access sequentially or even randomly.

**How can we run batch mode in MySQL?**

To perform batch mode in MySQL, we use the following command:

1. mysql;
2. mysql mysql.out;

### What are federated tables?

Federated tables are tables that point to the tables located on other databases on some other server.

### What is the difference between primary key and candidate key?

To identify each row of a table, we will use a primary key. For a table, there exists only one primary key.

A candidate key is a column or a set of columns, which can be used to uniquely identify any record in the database without having to reference any other data.

### What are the drivers in MySQL?

Following are the drivers available in MySQL:

* PHP Driver
* JDBC Driver
* ODBC Driver
* C WRAPPER
* PYTHON Driver
* PERL Driver
* RUBY Driver
* CAP11PHP Driver
* Ado.net5.mxz

## MySQL Advance Query

### MYSQL COMMAND LINE COMMANDS

| **COMMAND** | **MEANING** | **SYNTAX** |
| --- | --- | --- |
| mysql | Allows user to connect to the MySQL CLI | >MYSQL -U [USERNAME] -P; |
| exit | Exits the MySQL CLI | >EXIT; |
| clear | Clears the MySQL shell | >SYSTEM CLEAR; |
| create user | Creates a new user | >CREATE USER `NEWUSER`@`LOCALHOST` IDENTIFIED BY `NEW\_PASSWORD` |
| show user | Shows all user who have access to the MySQL Client | >SELECT USER, HOST FROM MYSQL.USER; |
| drop user | To delete an existing user | > DROP USER 'USERNAME'@'LOCALHOST'; |
| grant all privileges | Assigns privileges to a MySQL user | >GRANT ALL PRIVILEGES ON \* . \* TO 'USERNAME'@'LOCALHOST'; |
| show grants | Displays the privileges that are assigned to a MySQL user | > SHOW GRANTS FOR 'USERNAME'@'LOCALHOST'; |
| revoke all privileges | Revokes all privileges assigned to a MySQL user | >REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'USERNAME'@'LOCALHOST'; |
| mysqldump | Creates a backup of a set of SQL statements that can be used to recreate the original database object definitions and table data. | >MYSQLDUMP -U USERNAME -P DATABASENAME> DATABASENAME\_BACKUP.SQL |

### MYSQL DATABASE COMMANDS (DATA DEFINITION LANGUAGE;DDL)

| **COMMAND** | **MEANING** | **SYNTAX** |
| --- | --- | --- |
| show database | Shows all the databases available in MySQL server. | >SHOW DATABASE; |
| create database | Creates a new database if it does not exist. | >CREATE DATABASE DATABASENAME; |
| drop database | To delete an existing database permanently. | >DROP DATABASE  DATABASE\_NAME |
| alter database | Changes or modifies the characteristics of an existing database. | >ALTER DATABASE [DATABASENAME] ALTEROPTION ; |
| use database | Allow you to use a particular database or change from the current database to another database. | >USE DATABASENAME; |

### MySQL Table commands(DDL)

| **COMMAND** | **MEANING** | **SYNTAX** |
| --- | --- | --- |
| show tables | Shows all tables within the current database. | >SHOW TABLES; |
| create table | Creates a new table in the current database. | >CREATE TABLE TABLENAME (     COLUMN1 DATATYPE,     COLUMN2 DATATYPE,     COLUMN3 DATATYPE,    ....  CONSTRAINTS ....  ); |
| alter table (add column) | Adds a new column to an existing table. | >ALTER TABLE TABLENAME ADD COLUMNNAME DATATYPE; |
| alter table (drop column) | Deletes a column from an existing table. | >ALTER TABLE TABLENAME DROP COLUMN COLUMNNAME; |
| alter table (alter column) | Alters an existing column in an already existing table. | >ALTER TABLE TABLENAME  ALTER COLUMN COLUMNNAME DATATYPE; |
| alter table(add primary key) | Alters or adds primary key to an existing table. | >ALTER TABLE TABLENAME  ADD PRIMARY KEY (COLUMNNAME,...); |
| alter table(drop primary key) | Drops an existing primary key in a table. | >ALTER TABLE TABLENAME  DROP PRIMARY KEY; |
| alter table(add foreign key) | Creates a foreign key on an existing table. | >ALTER TABLE TABLENAME1 ADD FOREIGN KEY (COLUMN1) REFERENCES TABLENAME2(COLUMN2); |
| alter table(drop foreign key) | Deletes an existing foreign key in an already existing table. | > ALTER TABLE TABLENAME DROP FOREIGN KEY FOREIGNKEY\_NAME; |
| rename table | Changes the name of an existing table. | >RENAME TABLE OLD\_TABLENAME TO NEW\_TABLENAME; |
| drop table | Deletes the entire table along with its definition. | >DROP TABLE TABLE\_NAME; |
| truncate table | Remove all records in a MySQL table. | >TRUNCATE TABLE TABLENAME; |
| describe table | Displays all the columns of an existing table. | >DESCRIBE TABLE\_NAME; |
| describe table column | Displays all the values stored in a particular column. | >DESCRIBE TABLE\_NAME COLUMN\_NAME; |

### MySQL DML(Data Manipulation Language) Commands

| **COMMAND** | **MEANING** | **SYNTAX** |
| --- | --- | --- |
| select \* | Displays all rows in a table. | >SELECT \* FROM TABLENAME |
| select \* (multiple tables) | Displays all the rows of the cartesian product of the two tables | >SELECT \* FROM TABLENAME1,TABLENAME2; |
| select columns | Select particular columns from table(s) | >SELECT COLUMN1,COLUMN2 FROM TABLENAME; |
| select with condition | Displays rows based on a particular condition | > SELECT \* FROM TABLENAME WHERE CONDITION |
| select with multiple conditions(AND) | Displays rows only when both the conditions are satisfied. | > SELECT \* FROM TABLENAME WHERE CONDITION1 AND CONDITION2. |
| select with multiple conditions(OR) | Displays rows only when either of the conditions are satisfied. | > SELECT \* FROM TABLENAME WHERE CONDITION1 OR CONDITION2. |
| select with condition(NOT) | Displays rows based on negation of a particular condition. | >SELECT \* FROM TABLENAME WHERE NOT CONDITION. |
| select with group by | Displays rows that have same values into summary rows | > SELECT .. FROM .. WHERE… GROUP BY COLUMN3; |
| select with having | Used instead of where for aggregate functions. | >SELECT COUNT(COLUMN1) FROM TABLENAME ORDER BY COLUMN2 HAVING COUNT(COLUMN1)>3; |
| select distinct | Display all unique rows discarding duplicate ones. | >SELECT DISTINCT (COLUMN1) FROM TABLENAME; |
| order by | Used to sort results in ascending order or descending order | > SELECT … FROM TABLENAME ORDER BY COLUMN1 ASC|DESC; |
| column alias | Changes the output of the name of the column. | > SELECT COLUMN1 AS NEWNAME FROM TABLENAME; |
| like | Used to search for a specific pattern. | > SELECT COLUMN1 FROM TABLENAME WHERE COLUMN1 LIKE ‘%PATTERN%’; |
| insert record | Adds a new row to an existing table. | > INSERT INTO TABLENAME (COLUMN1,COLUMN2…) VALUES (VALUE1,VALUE2…); |
| insert record(multiple) | Adds multiple records into an existing table. | > INSERT INTO TABLENAME (COLUMN1,COLUMN2…) VALUES (VALUE1A,VALUE2A…),(VALUE1B,VALUE2B,...); |
| delete | Deletes all records in a table. | > DELETE FROM TABLENAME; |
| delete with where | Deletes specific records | >DELETE FROM TABLENAME WHERE CONDITION; |
| between | Selects values in a given range | >SELECT \* FROM TABLENAME WHERE AGE BETWEEN 25 AND 30. |
| in | Used instead of multiple OR operators. | > SELECT \* FROM TABLENAME WHERE COLUMN2 IN (V1,V2…); |
| exists | Tests for existence of a certain record. Returns a boolean value. | > SELECT \* FROM TABLE NAME WHERE EXIST (SUB QUERY); |
| update table | Modifies data in existing tables. | > UPDATE TABLENAME SET COLUMNNAME=VALUE WHERE CONDITION; |
| inner join | Selects records that have the same values in two same or distinct tables. | > SELECT COLUMN(S) FROM TABLENAME1 INNER JOIN TABLENAME2 ON TABLENAME1.COLUMNAME=TABLENAME2.COLUMNAME; |
| left join | Selects all the records from the left table and matching records from the right table. | >SELECT COLUMN(S) FROM TABLENAME1 LEFT JOIN TABLENAME2 ON TABLENAME1.COLUMNAME=TABLENAME2.COLUMNAME; |
| right join | Selects all the records from the right table and matching records from the left table. | >SELECT COLUMN(S) FROM TABLENAME1 RIGHT JOIN TABLENAME2 ON TABLENAME1.COLUMNAME=TABLENAME2.COLUMNAME; |
| cross join | Selects rows from cartesian product of both the tables. | >SELECT COLUMN(S) FROM TABLE1 CROSS JOIN TABLE2; |
| full outer join | Selects all records with a match on table1 or table2. | >SELECT COLUMN(S) FROM TABLENAME1 FULL OUTER JOIN TABLENAME2 ON TABLENAME1.COLUMNAME=TABLENAME2.COLUMNAME WHERE CONDITION; |
| union | Combines the result of two select statements. | >SELECT \* FROM TABLENAME1  UNION SELECT \* FROM TABLENAME2 |
| union all | Similar to Union but allows duplicate values | >SELECT \* FROM TABLENAME1  UNION ALL SELECT \* FROM TABLENAME2 |
| concat() | Combines two or more columns together. | >SELECT CONCAT(COLUMN1, " ", POSTALCODE, " ", COLUMN2) AS NEWCOL  FROM TABLENAME; |

### MySQL DATA TYPES

In MySQL just like other programming languages, each column, local variable, expression, and parameter has a related data type. A data type is an attribute that specifies the type of data that the object can hold.

**String Data Types**

| **DATATYPE** | **DETAILS** |
| --- | --- |
| CHAR(size) | Stores Alpha Numeric and special characters. Size varies from 0 to 255 characters. |
| VARCHAR(size) | Can contain letters, numbers, and characters that are of variable length (size). The size parameter specifies the column length in characters; it can be from 0 to 65535. |
| BINARY(size) | Similar to CHAR(). But it stores binary strings. |
| VARBINARY(size) | Similar to Binary() but the length is variable. |
| TINYBLOB | For Binary Large Objects. Max size=255 bytes. |
| TINYTEXT | Holds string of max length 255 characters. |
| TEXT(Size) | Stores a string of max length 65535 bytes. |
| BLOB | Stores Binary Large Objects up to 65535 bytes of data. |
| MEDIUMTEXT | Stores 2^8 times the characters as compared to TINYTEXT. |
| MEDIUMBLOB | Stores 2^8 times bytes as compared to TINYBLOB. |
| LONGTEXT | Stores 2^8 times the characters as compared to MEDIUMTEXT. |
| LONGBLOB | Stores 2^8 times bytes as compared to MEDIUMBLOB. |
| ENUM(val1, val2, val3, …) | Stores only one value, which can be chosen from a range of possible values. An ENUM list can contain at most 65535 values. A value that is inserted that is not in the list will be replaced with a blank value. The values are arranged in the order you specify them. |
| SET(val1, val2, val3, …) | Stores a string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list |

**Numeric Data Types**

| **DATATYPE** | **DETAILS** |
| --- | --- |
| BIT(size) | Stores a bit-value. The size parameter specifies the number of bits per value . The value is represented as a number of bits. The size parameter can hold a value from 1 to 64. The default value for size is 1. |
| TINYINT(size) | Stores very small int values. Signed ranges  from -128 to 127. Unsigned ranges from 0 to 255. Size defines the maximum display width of 255. |
| BOOL | Zero is considered as false and one is considered as true. |
| BOOLEAN | Same as BOOL. |
| SMALLINT(size) | Stores a small integer. Signed ranges from -32768 to 32767. Unsigned ranges from 0 to 65535. Size defines the maximum display width of 255. |
| MEDIUMINT(size) | Stores a medium valued integer. Signed ranges from -8388608 to 8388607. Unsigned ranges from 0 to 16777215. Size defines the maximum display width of 255. |
| INT(size) | Stores a medium integer. Signed ranges from -2147483648 to 2147483647. Unsigned ranges from 0 to 4294967295. Size defines the maximum display width of 255. |
| INTEGER(size) | Same as INT(size) |
| BIGINT(size) | Stores a large valued integer. Signed ranges  from -9223372036854775808 to 9223372036854775807. Unsigned ranges  from 0 to 18446744073709551615. Size defines the maximum display width of 255. |
| FLOAT(size, d) | Stores a floating point(decimal number). The number of digits is specified in size. The number of digits after the decimal point is specified by the value d. |
| FLOAT(p) | Stores a floating point(decimal number. If p value is between 0 and 24, the data type becomes FLOAT() else the data type becomes DOUBLE() |
| DOUBLE(size, d) | Stores a  normal-size floating point (decimal)number. The number of digits is specified in size. The number of digits after the decimal point is specified by the value d. |
| DECIMAL(size, d) | An exact fixed-point number. The total number of digits is specified in size. The number of digits after the decimal point is specified in the d parameter. The maximum number for size is 65. The maximum number for d is 30. The default value for size is 10. The default value for d is 0. |

**Date and Time Data Types**

| **DATATYPE** | **DETAILS** |
| --- | --- |
| DATE | Stores a date in the format: YYYY-MM-DD. Supports a range between ‘1000-01-01’ to ‘9999-12-31’ |
| DATETIME(fsp) | Combination of date and time in the format: YYYY-MM-DD hh:mm:ss. Supports a range between ‘1000-01-01 00:00:00’ to ‘9999-12-31 23:59:59’. |
| TIMESTAMP(fsp) | Stores a time stamp in the format YYYY-MM-DD hh:mm:ss UTC. Supports a range between ‘1970-01-01 00:00:01’ UTC to ‘2038-01-09 03:14:07’ UTC. |
| TIME(fsp) | Stores time in the format hh:mm:ss. Supports a range between ‘-838:59:59’ to ‘838:59:59’ |
| YEAR | Stores a year in four-digit format. Supports a range between 1901 to 2155 (includes 0000). |

### MySQL AGGREGATE FUNCTIONS

A function that performs an arithmetic operation on a set of values and returns a single value is called an aggregate function.

| **COMMAND** | **FUNCTION** | **SYNTAX** |
| --- | --- | --- |
| count() | Returns the number of rows, (including NULL) | >SELECT COUNT(COLUMN\_NAME)  FROM TABLE\_NAME  WHERE CONDITION; |
| sum() | Returns sum of all non NULL values. | >SELECT SUM(COLUMN\_NAME)  FROM TABLE\_NAME  WHERE CONDITION; |
| avg() | Returns average of all non NULL values. | >SELECT AVG(COLUMN\_NAME)  FROM TABLE\_NAME  WHERE CONDITION; |
| min() | Returns minimum value in the set. | >SELECT MIN(COLUMN\_NAME)  FROM TABLE\_NAME  WHERE CONDITION; |
| max() | Returns maximum value in the set. | >SELECT MAX(COLUMN\_NAME)  FROM TABLE\_NAME  WHERE CONDITION; |
| groutp\_concat() | Concatenates values from multiple rows into one field. | >SELECT COLUMN1, COLUMN2, ...  GROUP\_CONCAT ( DISTINCTCOLUMN1  ORDER BY .. )  FROM TABLE\_NAME GROUP BY COLUMN2; |

### INDEXES AND VIEWS IN MySQL

An Index retrieves data much faster than otherwise. Indexes speed up the query/search. A user cannot view an Index. Updating a table with an index takes more time because both table and index have to be updated.

The view is a virtual table which takes the result of an SQL query. Users can access a View. They have rows and columns similar to a table.

| **COMMAND** | **FUNCTION** | **SYNTAX** |
| --- | --- | --- |
| create index | Creates a new index from an existing table. Allows duplicate values. | > CREATE INDEX indexname  ON tablename (column1, column2, ...); |
| create index unique | Similar to creating an index. But only allows unique values. | >CREATE UNIQUE INDEX indexname  ON tablename (column1, column2, ...); |
| drop index | Deletes an existing index. | > DROP INDEX INDEXNAME; |
| rebuild index | Used to rebuild one or all indexes in a table if corrupted. | >REINDEX INDEX INDEXNAME; |
| create view | Creates a view if it doesn’t exist. | > CREATE VIEW VIEWNAME AS SELECT COLUMN1,COLUMN2 FROM TABLE WHERE CONDITION; |
| update view | Creates or edits an existing view. | > CREATE OR REPLACE viewname  AS  SELECT COLUMN1,COLUMN2 FROM TABLE WHERE CONDITION; |
| rename view | Changes the name of the view. | > RENAME TABLE VIEWNAME TO NEWVIEWNAME; |
| drop view | Deletes an existing view. | > DROP VIEW VIEWNAME; |
| drop views | Deletes multiple views. | > DROP VIEW VIEW1,VIEW2…; |
| show views | Displays all views in a database. | > SHOW FULL TABLES  [{FROM | IN } databasename]  WHERE table\_type = 'VIEW'; |

### TRIGGERS IN MYSQL

Triggers are DBMS objects which are associated with tables. Triggers are fired when any one of the DML statements (INSERT, DELETE or UPDATE) is activated.

There are two types of triggers,

* Row Level Triggers: A trigger is an instruction that causes a row to trigger to be fired once for each row affected by an insert, update, or delete statement. The row trigger is fired automatically.
* Statement Level Trigger: Trigger is fired once regardless of the number of DML statements.

**There are six types of triggers, namely,**

* Before Insert: Activated before insertion.
* After Insert: Activated after insertion.
* Before Update: Activated before updating.
* After Update: Activated after updating.
* Before Delete: Activated before deletion.
* After Delete: Activated after deletion.

| **COMMAND** | **FUNCTION** | **SYNTAX** |
| --- | --- | --- |
| create trigger | Creates a new trigger on an existing table. | >CREATE TRIGGER TRIGGERNAME  BEFORE | AFTER INSERT | UPDATE| DELETE  ON TABLENAME FOR EACH ROW  TRIGGERBODY; |
| drop trigger | Deletes an existing trigger. | > DROP TRIGGER TRIGGERNAME; |
| show all triggers | Displays all the triggers in the database. | > SHOW TRIGGERS FROM | IN DATABASE\_NAME WHERE SEARCH\_CONDITION; |

### STORED PROCEDURES AND FUNCTION

Procedures are reusable SQL codes that we store in a database. We can directly call procedures instead of writing the query again and again.

Functions are reusable code, which runs certain SQL commands and returns an appropriate value.

**Syntax to create a new procedure.**

DELIMITER $

CREATE PROCEDURE procedurename(parameterlist)

BEGIN

body;

END $

DELIMITER ;

**Syntax to create a new function**

DELIMITER $

CREATE FUNCTION functionname(parameterlist)

RETURNS datatype

NOT DETERMINISTIC

BEGIN

%statements%

END $

DELIMITER ;

| **COMMAND** | **FUNCTION** | **SYNTAX** |
| --- | --- | --- |
| drop procedure | Deletes an existing procedure. | > DROP PROCEDURE PROCEDURENAME; |
| show all procedures | Displays all the stored procedures in the database. | > SHOW PROCEDURE STATUS LIKE ‘%PATTERN’ | WHERE CONDITION; |
| drop function | Deletes an existing stored function. | > DROP FUNCTION FUNCTIONNAME; |
| show stored functions | Displays all the stored functions. | > SHOW FUNCTION STATUS LIKE ‘%PATTERN’ | WHERE CONDITION; |

### INBUILT FUNCTIONS IN MySQL

**STRING FUNCTIONS**

| **Function** | **Description** |
| --- | --- |
| ASCII | Returns the ASCII value of a character |
| CHAR\_LENGTH | Returns the length of a string. |
| CHARACTER\_LENGTH | Returns the length of a string |
| CONCAT | Concatenates two or more expressions. |
| CONCAT\_WS | Concatenates with a separator. |
| FIELD | Returns the index of value in a list. |
| FIND\_IN\_SET | Returns the index of a string within a list. |
| FORMAT | Changes the format/representation. |
| INSERT | Inserts a string within a string at a given index. |
| INSTR | Returns the index of the first occurrence of a string in another one. |
| LCASE | Converts an entire string to lowercase. |
| LEFT | Extracts a length of characters from the left of a string. |
| LENGTH | Returns the string length in bytes. |
| LOCATE | Returns the location of the first occurrence of a substring in a given  string |
| LOWER | Converts an entire string to lowercase. |
| LPAD | Left-pads a string with a given string. |
| LTRIM | Removes spaces from the left of a string. |
| MID | Extracts a substring from a string at a given position. |
| POSITION | Returns the location of the first occurrence of a substring in a given  string |
| REPEAT | Repeats the string the number of times the user specifies. |
| REPLACE | Replaces occurrences of a substring in a string with another substring. |
| REVERSE | Reverses the string. |
| RIGHT | Extracts a length of characters from the right of a string. |
| RPAD | Right-pads a string with a given string. |
| RTRIM | Removes spaces from the right of a string. |
| STRCMP | Checks whether two strings are equal. |
| SUBSTR | Extracts a substring from a string at a position mentioned by the user. |
| SUBSTRING | Same as substr. |
| TRIM | Trims leading and trailing spaces from a string as specified by the user. |
| UCASE | Converts an entire string to uppercase. |
| UPPER | Converts an entire string to uppercase. |

**NUMERIC FUNCTIONS**

| **Function** | **Description** |
| --- | --- |
| ABS | Returns the absolute value. |
| ACOS | Returns the cosine inverse. |
| ASIN | Returns the sine inverse. |
| ATAN | Returns the tan inverse of one or two numbers. |
| ATAN2 | Returns the tan inverse of  two numbers. |
| AVG | Returns the mean value. |
| CEIL | Returns the smallest integer that is greater than or equal to the number |
| CEILING | Returns the smallest integer that is greater than or equal to the number |
| COS | Returns the cosine. |
| COT | Returns the cotangent. |
| COUNT | Returns the number of records returned by a query. |
| DEGREES | Converts angle in Radians to Degrees. |
| DIV | Integer division |
| EXP | Returns e raised to the power of value mentioned. |
| FLOOR | Returns the largest integer that is less than or equal to a number |
| GREATEST | Returns the largest value in the list. |
| LEAST | Returns the smallest value in the list. |
| [L](https://www.w3schools.com/mysql/func_mysql_ln.asp)N | Calculates logarithm to the base e. |
| LOG | Calculates logarithm to the base e. |
| LOG10 | Calculates logarithm to the base 10. |
| LOG2 | Calculates logarithm to the base 2. |
| MAX | Returns the largest value in a set. |
| MIN | Returns the least value in a set. |
| MOD | Returns the remainder after division of two numbers. |
| PI | Returns value of **π** |
| POW | Used for exponents. |
| POWER | Used for exponents. |
| RADIANS | Converts angle in Degree to Radians. |
| RAND | Generates a random number. |
| ROUND | Rounds the number to the nearst decimal place. |
| SIGN | Returns the sign of a number |
| SIN | Returns the sine. |
| SQRT | Returns the root of a number. |
| SUM | Calculates the sum of a set. |
| TAN | Returns the tangent. |

**MYSQL DATE FUNCTION**

| **Function** | **Description** |
| --- | --- |
| ADDDATE | Adds a date interval and return the value. |
| ADDTIME | Adds a time interval and then returns the value. |
| CURDATE | Returns today’s date |
| CURRENT\_DATE | Same as CURDATE |
| CURRENT\_TIME | Returns the time at the moment |
| CURRENT\_TIMESTAMP | Returns date and time at the moment. |
| CURTIME | Returns time at the moment. |
| DATE | Picks up the date from an expression of Date/Time. |
| DATEDIFF | Returns number of days between two given dates. |
| DATE\_ADD | Similar to ADDDATE |
| DATE\_FORMAT | Changes the format in which Date is displayed. |
| DATE\_SUB | Subtracts a time interval and returns the value. |
| DAY | Returns the weekday for today. |
| DAYNAME | Returns the weekday name for any date. |
| DAYOFMONTH | Used to retrieve the index of the day of the month of any date. |
| DAYOFWEEK | Used to retrieve the index of the weekday of any date. |
| DAYOFYEAR | Used to retrieve the index of the day of a year of any date. |
| EXTRACT | Extracts a part of any date. |
| HOUR | Returns the “hours” in a given time. |
| LAST\_DAY | Return the last day of the given month. |
| LOCALTIME | Returns the date and time at the moment. |
| LOCALTIMESTAMP | Similar to LOCALTIME. |
| MAKEDATE | Returns a date based on the year and the no. of days you specify. |
| MAKETIME | Returns a time based on the hours , minutes and seconds you specify. |
| MICROSECOND | Returns the microseconds in a given time. |
| MINUTE | Returns the minutes in a given time. |
| MONTH | Returns the month on a given date. |
| MONTHNAME | Same as MONTH but returns the name of the month. |
| NOW | Returns date and time at the moment. |
| PERIOD\_ADD | Adds a specific number of months. |
| PERIOD\_DIFF | Return the difference between two time periods. |
| SECOND | Return the seconds in a given time. |
| SEC\_TO\_TIME | Returns time in seconds. |
| STR\_TO\_DATE | Formats the date based on a particular string. |
| SUBDATE | Same as DATE\_SUB. |
| SUBTIME | Subtracts a time interval. |
| SYSDATE | Returns the date/time reflected by the system. |
| TIME | Returns the time from a date/time value. |
| TIME\_FORMAT | Time is displayed based on a certain format. |
| TIME\_TO\_SEC | Returns time in seconds. |
| TIMEDIFF | Returns the difference between two date-time values. |
| TO\_DAYS | Returns the number of days between amy date and “0000-00-00” |

**ADVANCED MYSQL FUNCTION**

| **Function** | **Description** |
| --- | --- |
| BIN | Returns binary value of a given number. |
| BINARY | Converts a given string to a binary string. |
| CAST | Converts data from one data type to another. |
| COALESCE | Returns the first non-null value in a set or list. |
| CONV | Converts a number from one number-base system to another |
| CONVERT | Similar to CAST in working |
| CURRENT\_USER | Returns the user name and host name for the MySQL account that is currently used. |
| DATABASE | Returns the name of the database currently in use. |
| IF | IF condition statement. |
| SESSION\_USER | Returns the current MySQL user name and host name. |
| SYSTEM\_USER | Similar to SESSION\_USER. |
| USER | Similar to SESSION\_USER. |
| VERSION | Returns the current version of the MySQL server installed. |

