

PHP NOTES

❖ What is PHP?

- **PHP stands** for PHP: Hypertext Preprocessor.
- **PHP** is a server scripting language, it's free to download & use.
- **PHP** is a powerful tool for making dynamic and interactive Web pages.
- **PHP** scripts are executed on the server.
- **PHP** can create, open, read, write, delete, and close files on the server.
- **PHP** can collect form data, send and receive cookies.
- **PHP** can add, delete, modify data in your database.

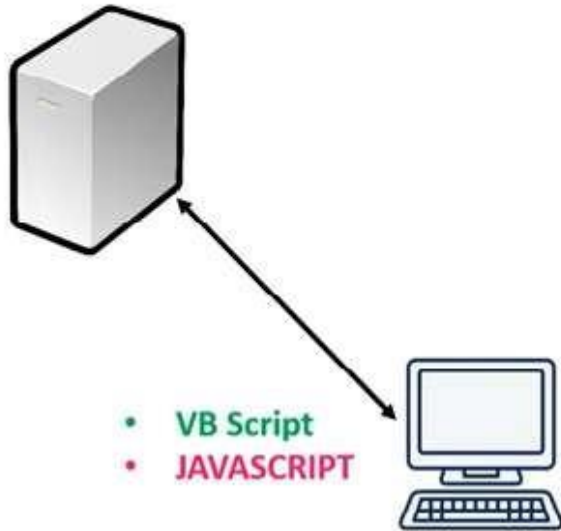
Before you start studying PHP, you should have a basic knowledge of:

- i. Html
- ii. Css
- iii. Javascript

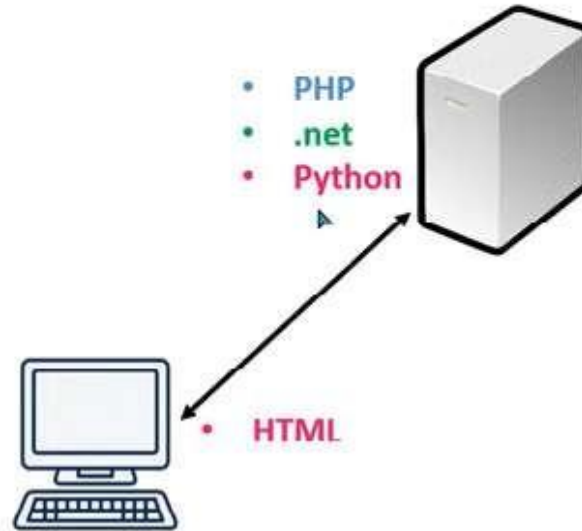


Web based Scripting Language

Client Side Script



Server Side Script



- Create Dynamic website with database (static and synamic website).
- Web based software application

❖ What is a PHP File?

PHP files can contain text, HTML, CSS, JavaScript, and PHP code.

PHP code is executed on the server, and the result is returned to the browser as plain HTML.

PHP files have extension “.php”.

Since **PHP** is a server-side (back-end) scripting language, the code written in the PHP file is executed on the server.

Often when you fill out a form online or submit contact details to a website the backend code will send that information to a server using a script inside the PHP file.

❖ What Can PHP Do?

PHP files can contain text, HTML, CSS, JavaScript, and PHP code.

PHP can generate dynamic page content.

PHP can create, open, read, write, delete, and close files on the server.

PHP can collect form data.

PHP can add, delete, modify data in your database.

❖ Why PHP?

PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.).PHP is compatible with almost all servers used today (Apache, IIS, etc.).

PHP is easy to learn and runs efficiently on the server side

Set Up PHP on Your Own PC

However, if your server does not support PHP, you must:

- install a web server
- install PHP
- install a database, such as MySQL
- Or Install Wamp/Xampp server(3rd party sw)
,browser,editor <http://www.wampserver.com/en/>

Basic PHP Syntax :-

A PHP script can be placed anywhere in the document.

A PHP script starts with `<?php` and ends with `?>`:

A PHP file normally contains HTML tags, and some PHP scripting code.

The default file extension for PHP files is ".php". Example

```
<?php // PHP code goes here ?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My first PHP page</h1>
```

```
<?php echo "Hello World!"; ?>
```

```
</body>
```

```
</html>
```

❖ PHP Variables:-

In PHP, a variable starts with the \$ sign, followed by the name of the variable:

Example

```
$x = 5;  
$y = "Reva Tech";
```

The PHP echo Statement

The echo statement can be used to get output

Display Variables

The following example shows how to output text and variables

```
<?php  
$txt = "Learn PHP";  
echo "<h2>" . $txt . "</h2>";  
?>
```

In the example above, the variable \$x will hold the value 5, and the variable \$y will hold the value " RevaTech .

❖ PHP Conditional Statements:-

In PHP we have the following conditional statements:

if statement - executes some code if one condition is true if...else statement - executes some code if a condition is true and another code if that condition is false

if...elseif...else statement - executes different codes for more than two conditions.

- **PHP If Statement**

PHP if statement allows conditional execution of code. It is executed if condition is true.

If statement is used to execute the block of code exist inside the if statement only if the specified condition is true.

Syntax

```
if(condition){  
    //code to be executed.
```

Example: -

```
<?php
$num=12;
if($num<100){
echo "$num is less than 100";
}
?>
```

Output: -

12 is less than 100

- **PHP If-else Statement**

PHP if-else statement is executed whether condition is true or false.

It executes one block of code if the specified condition is **true** and another block of code if the condition is **false**.

Syntax: -

```
if(condition){
//code to be executed if true
}else{
```



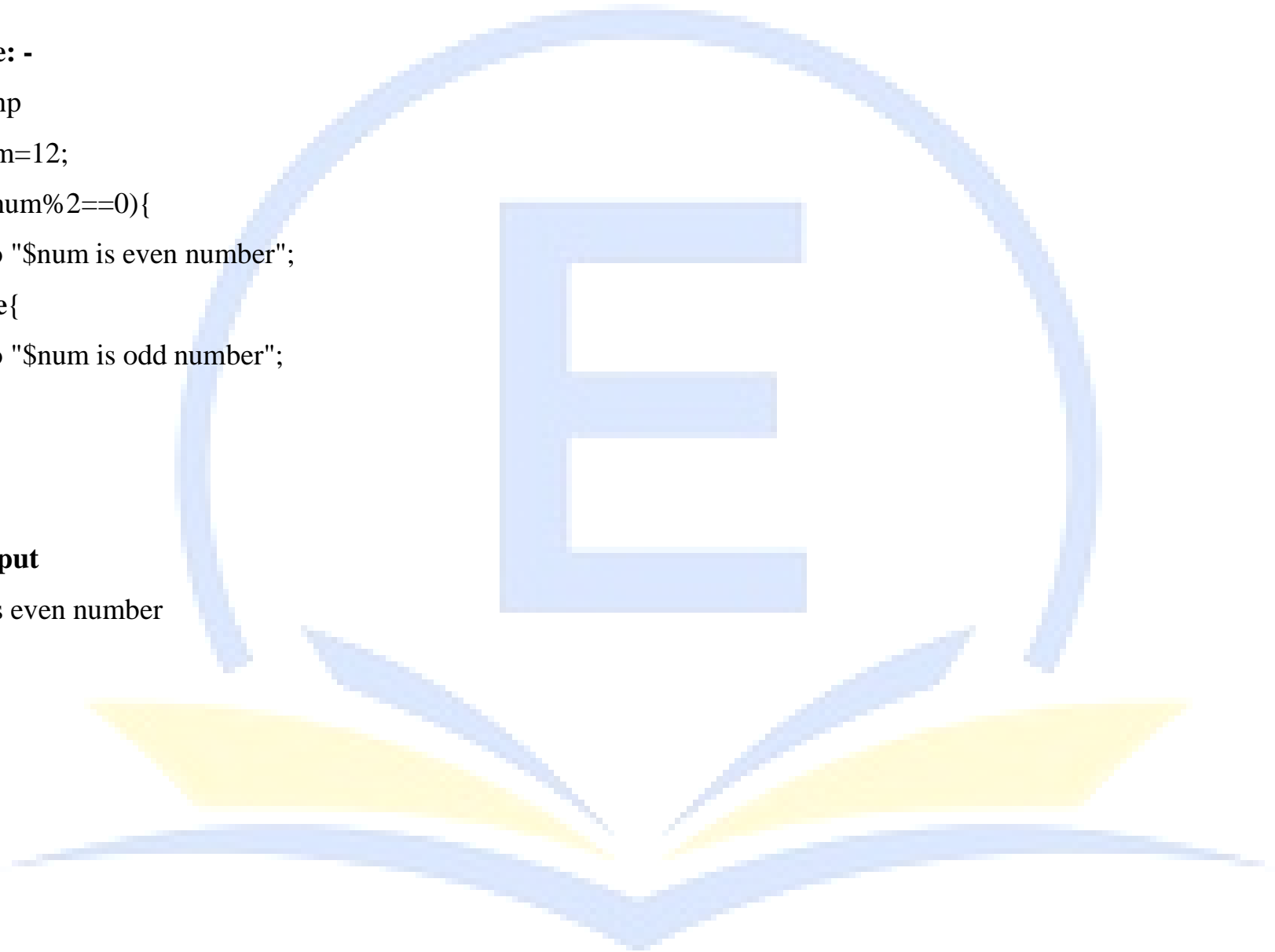
```
//code to be executed if false  
}
```

Example: -

```
<?php  
$num=12;  
if($num%2==0){  
echo "$num is even number";  
}else{  
echo "$num is odd number";  
}  
?>
```

Output

12 is even number



❖ PHP - The if...elseif...else Statement: -

The if...elseif...else statement executes different codes for more than two conditions.

Syntax

```
if (condition1){  
    //code to be executed if condition1 is true  
} elseif (condition2){  
    //code to be executed if condition2 is true  
} else{  
    //code to be executed if all given conditions are false  
}
```

Example: -

```
$t = date("H");  
if ($t < "10") {  
    echo "Have a good morning!";  
} elseif ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}
```

Output: -

Have a good day!

❖ PHP Loops: -

while - loops through a block of code as long as the specified condition is true

for - loops through a block of code a specified number of times foreach - loops through a block of code for each element in an array.

The PHP while Loop

The while loop executes a block of code as long as the specified condition is true.

Example:

```
$i = 1;
while ($i < 6) {echo $i;
  $i++;
}
```

Output: -

12345

- **break Statement**

With the break statement we can stop the loop even if the condition is still true.

Example: -

Stop the loop when \$i is 3

```
$i = 1; while ($i < 6) {if ($i == 3)
break; echo $i; $i++; }
```

❖ The PHP for Loop

The for loop - Loops through a block of code a specified number of times.

The for loop is used when you know how many times the script should run.

Syntax

```
for (expression1, expression2, expression3) {  
    // code block  
}
```

how it works?

expression1 is evaluated once

expression2 is evaluated before each iteration

expression3 is evaluated after each iteration

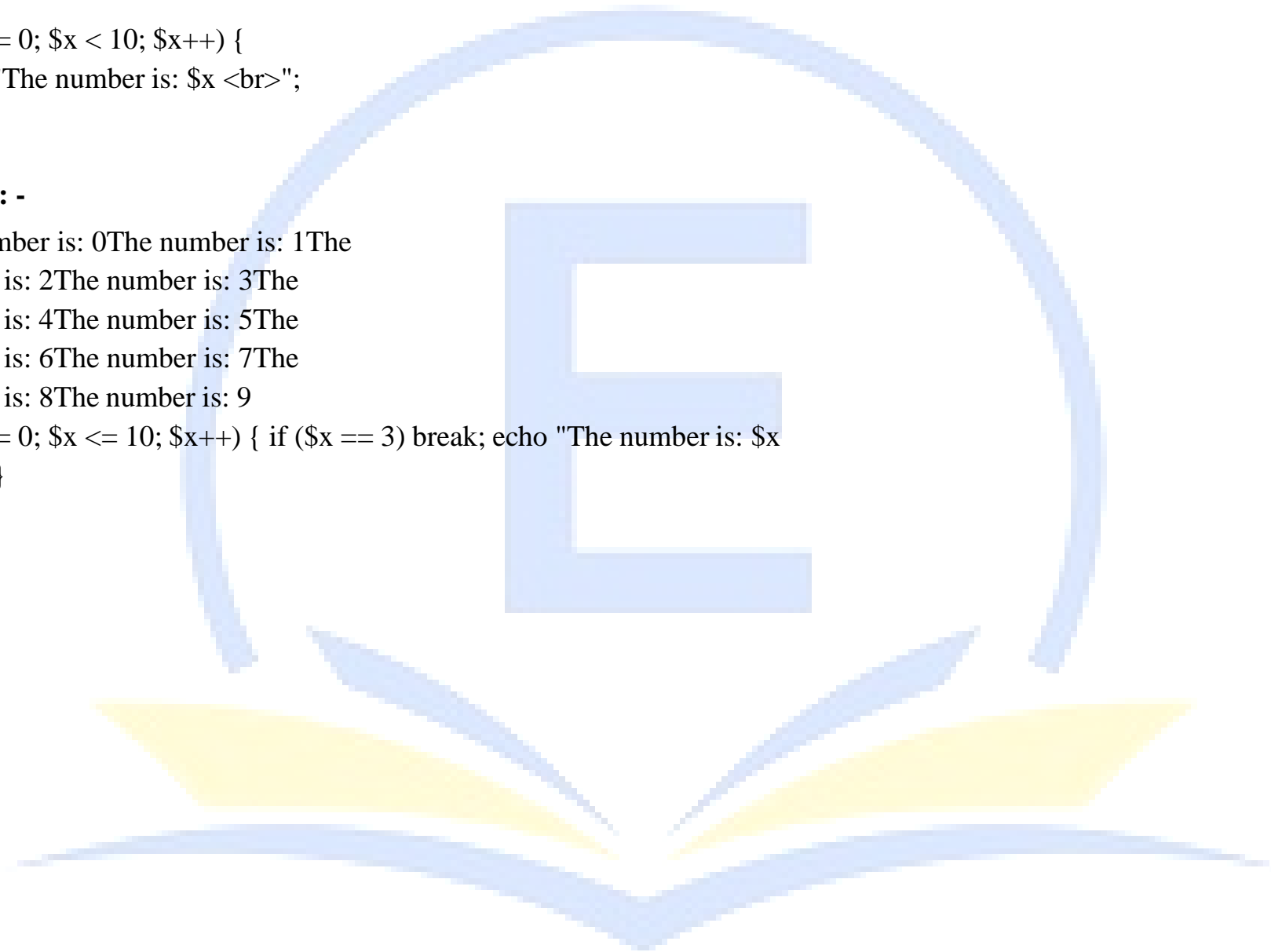
Example: -

```
<?php
for ($x = 0; $x < 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

Output: -

The number is: 0The number is: 1The
number is: 2The number is: 3The
number is: 4The number is: 5The
number is: 6The number is: 7The
number is: 8The number is: 9

```
for ($x = 0; $x <= 10; $x++) { if ($x == 3) break; echo "The number is: $x  
<br>"; }
```



❖ PHP foreach Loop: -

The most common use of the foreach loop, is to loop through the items of an array. Syntax

```
<?php
$colors = array("red", "green", "blue", "yellow");
foreach ($colors as $x) {
    echo "$x <br>";
}
?>
```

For every loop iteration, the value of the current array element is assigned to the variable \$x. The iteration continues until it reaches the last array element.

Output:

Red

Green

Blue

Yellow

blue

yellow

❖ Keys and Values: -

The array above is an indexed array, where the first item has the **key** 0, the second has the key 1, and so on.

Associative arrays are different, associative arrays use named keys that you assign to them, and when looping through associative arrays, you might want to keep the **key as well as the value**.

This can be done by specifying both the key and value in the foreach definition, like this:

```
<?php
$members = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach ($members as $x => $y) {echo "$x : $y <br>";
}
?>
```

Output: -

Peter : 35

Ben : 37

Joe : 43

❖ PHP Functions: -

A function is a block of statements that can be used repeatedly in a program

PHP has more than 1000 built-in functions, and in addition you can create your own custom functions.

A user-defined function declaration starts with the keyword function, followed by the name of the function:

Create a Function

Example: -

```
<?php  
function myMessage() { //Create a Function  
    echo "Hello world!";  
}  
myMessage();//Call a Function  
?>
```

Output: -

Hello world!

❖ Two Types of Functions: -

1. Built-in Functions
2. User defined Functions

Built-in Functions

PHP has over 1000 built-in functions that can be called directly, from within a script, to perform a specific task.

For Example:

`Array (); explode();`

User Defined Functions

create your own functions.

A function is a block of statements that can be used repeatedly in a program.

A function will not execute automatically when a page loads.

A function will be executed by a call to the function.

```
<?php
```

```
function familyName($fname) { echo "$fname Thank You.<br>";
```

```
}
```

```
familyName("Sadhana");
```

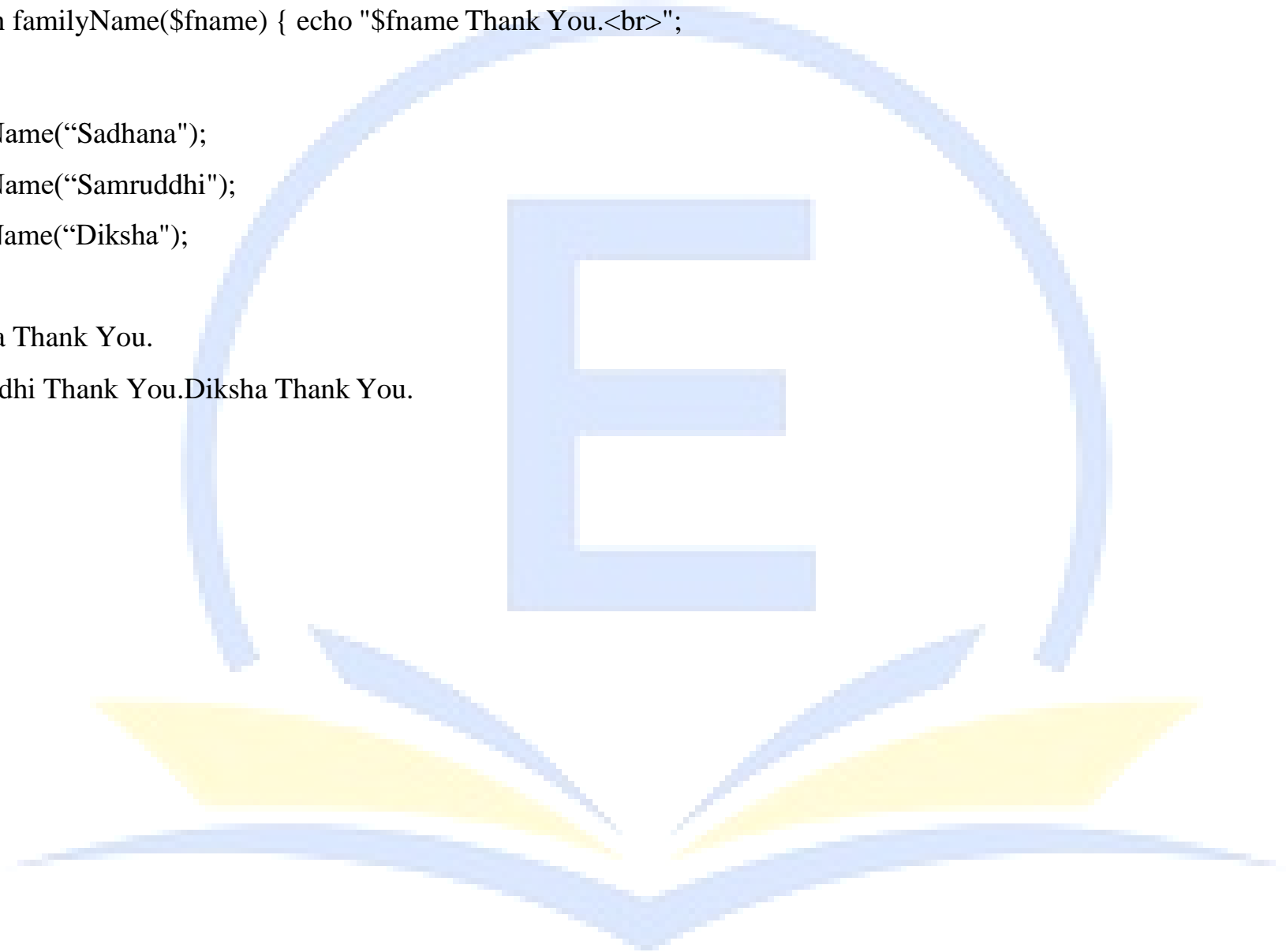
```
familyName("Samruddhi");
```

```
familyName("Diksha");
```

```
?>
```

Sadhana Thank You.

Samruddhi Thank You.Diksha Thank You.



The following example has a function with two arguments (\$fname, \$year):

```
<?php  
function familyName($fname, $year) {echo "$fname Born in $year <br>";  
}  
familyName("Sadhana","1992");familyName("Samruddhi","2023");  
familyName("Diksha","2019");  
?>
```

Priyanka Born in 1992
Reva Born in 2023
Vihaa Born in 2019

The PHP superglobals `$_GET` and `$_POST` are used to collect form-data. The form data is sent with the HTTP POST method.

The **`$_POST`** variable collects the data from the HTML form via the POST method. After submitting an HTML Form some data is generated which is collected using a specific method POST.

The **`$_GET`** variable collects the data from the url via the GET method.

❖ Mysql query/ functions: -

Isset() :

The **isset()** function checks whether a variable is set. **mysql_query** : Send query to MySQL database and execute. **mysql_connect** : This function is use to connect a database. **mysql_select_db**: This function sets the active MySQL database (returns TRUE on success, or FALSE on failure).

mysql_num_rows:

Retrieves the number of rows from a result set. **mysql_insert_id** : function returns the id (generated with AUTO_INCREMENT) from the last query.

mysql_fetch_array: function is used to fetch rows from the database and store them as an array.

array :

An **array** is a special variable that can hold many values under a single name, and you can access the values by referring to an index number or name.

The **print_r()** function prints the information about a variable in a more human-readable way.

The **sizeof()** function returns the number of elements in an array.

❖ Mysql Queries: -

select query : The SQL **SELECT** statement is used to select the records from databasetables.

Syntax : select * from table_name;

INSERT INTO statement is used to insert new rows in a database table.Syntax :

INSERT INTO TABLE_NAME (column1, column2, column3, ... columnN) VALUES (value1,value2, value3,

The **UPDATE** statement is used to update existing records in a table.

Syntax :

UPDATE table_name

SET column1=value, column2=value2,...

WHERE some_column=some_value

The **DELETE** statement is used to delete records from a table.Syntax :

DELETE FROM table_name

WHERE some_column = some_value

Mysql database connection in php :

```
<?php //connection.php
```

```
error_reporting('E_ALL');
```

```
session_start();
```

```
$servername = 'localhost';//server name
```

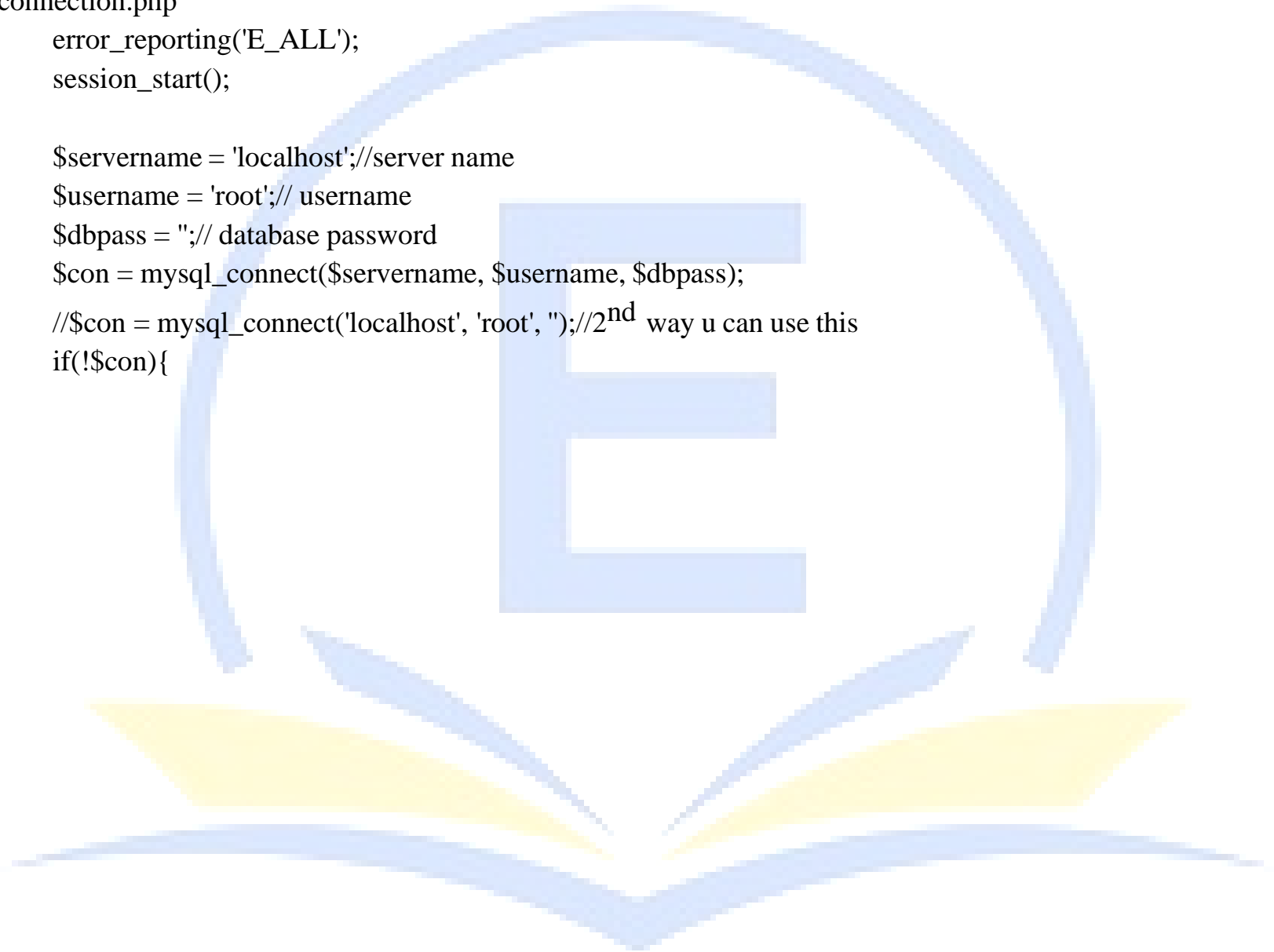
```
$username = 'root';// username
```

```
$dbpass = "";// database password
```

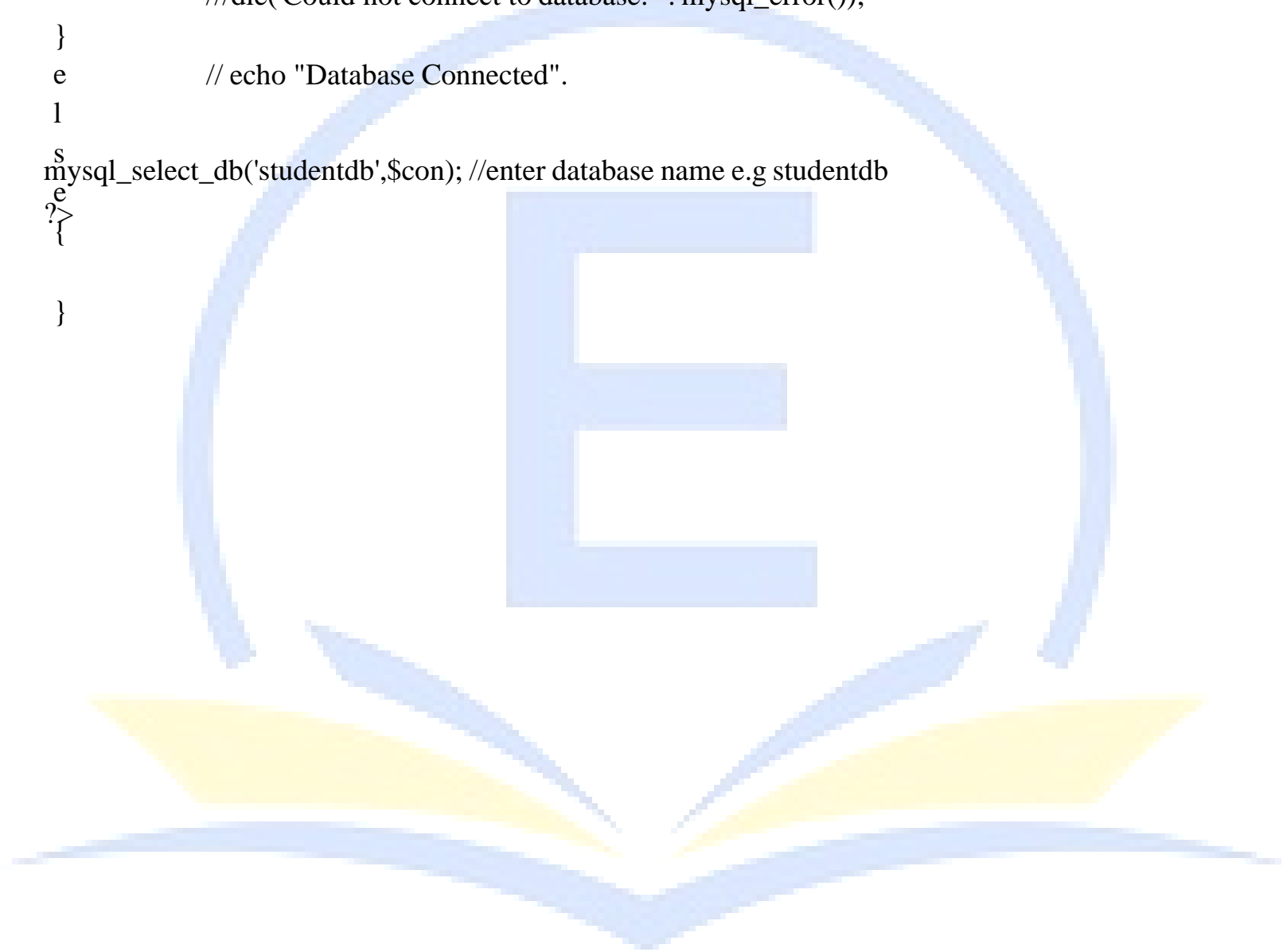
```
$con = mysql_connect($servername, $username, $dbpass);
```

```
// $con = mysql_connect('localhost', 'root', ''); // 2nd way u can use this
```

```
if(!$con){
```



```
echo "Database not Connected";  
    ///die('Could not connect to database: ' . mysql_error());  
}  
e    // echo "Database Connected".  
l  
s  
mysql_select_db('studentdb',$con); //enter database name e.g studentdb  
e  
?>  
{  
  
}
```



How to insert form record in database table using php : save file with name Form.php

Step 1: HTML Form

```
<div class="container" style="background-color:#ede9e9">
  <h2 class="col-lg-12 col-md-12 col-sm-12 col-xs-12 text-center " style="border-bottom:1px solid gray;padding-bottom:10px">Student
  Registration Form</h2>
  <form action="" method='post'>
    <div class='col-md-offset-1'>
      <div class="col-lg-5 col-md-5 col-sm-5 col-xs-3 text-left " >
        <label><b>Name</b></label>
      </div>
      <div class="col-lg-6 col-md-6 col-sm-6 col-xs-8 " style="margin-bottom:5px">
        <div class='col-md-6'>
          <input type="text" class='form-control input-sm' placeholder="Enter Student Name. ...."
          name="StudentName" id="StudentName" value="" tabindex="1" required
          style="border: 1px solid #68e136;" autofocus>
        </div>
      </div>
      <div class='col-md-offset-2' >
        <div class='col-md-8 text-center' style="margin-top:20px">
          <button type="submit" id="Save" name="save" class="btn btn-success">Submit</button>
          <button type="button" name="reset" class="btn btn-warning">Reset</button>
        </div>
      </div>
    </form>
  </div>
```


How to insert form record in database table using php : inForm.php file write below code

Step 2: include connection file or connection code in php file

```
<?php
// include file or write connection code here
include('connection.php');
?>
```

How to insert form record in database table using php : inForm.php file write below code

Step 3: write insert code in Form.php file

```
<?php if(isset($_POST['save'])) {
$StudentName = $_POST['StudentName'];

SqlStudent = "insert into tblstudent(Name) values('".$StudentName."')";
$ResStudent = mysql_query($SqlStudent);

echo "<script>alert('Student Added Successfully');window.location.assign('Form.php')</script>";
}
?>
```

❖ How to show form record from table using php mysql: -

```
<? php
$show="select * from tblstudent";
$showres=mysql_query($show);
?>
<table class="table table-bordered">
<tr> <th>Name</th></tr>
<?php
//used to fetch rows from the database and store them as an array.
while($row=mysql_fetch_array($showres))
{ ?>
<tr><td><?php echo $row['Name'];?></td></tr><?php } ?>
</table>
```

❖ How to login add Username and password in registration form and insert in table: -

```
<?php include('connection.php');
if(isset($_POST['UserLogin']))
{
    $UserName = $_POST['user'];//123 post textbox value
    $Password = $_POST['upwd'];//123
    // get details from table with using where condition
    $sql="select * from tbllogin where UserName='".$UserName.'" AND Password='".$Password.'" ";
    $res = mysql_query($sql);
    $rowcnt=mysql_num_rows($res);//get total number count for record found in table
    $row=mysql_fetch_array($res); // store tbllogin column value in $row array
    $StudentId=$row['StudentId'];// store student value in a vairable which is get from $row arrayif ( $rowcnt > 0) {//
    check record present in table then login
        $_SESSION['UserName']=$UserName; //123 store value in UserName Session variable
        $_SESSION['StudentId']=$StudentId;//23

    echo "<script>alert('Login Successfully. '); window.location.assign('StudentList.php');</script>";
    } else { echo "<script>alert('UserName & Password Not Matched');window.location.assign('login.php')</script>";
    }
}
?>
```