A complete guide to building web applications

Programming

PHP MYSQL

Let's Explore...

For advanced learning

Edition-3

Hirdesh Bhardwaj


BOOKS
CLINIC

Booksclinic Publishing

Website: - **www.booksclinic.com**
B.D. Complex, Near Tifra Over Bridge, Bilaspur, Chhattisgarh, India, 495001

ISBN: 978-93-90655-24-3
Book: Academic
First Edition Published by
Booksclinic Publishing 2021

Price: Rs.230/-

First Edition: **2016**

Third Edition: **2021**

Printed in India

PHP Mysql For Advanced Learning

(3rd Edition)

By

Hirdesh Bhardwaj

About the Author

Hirdesh Bhardwaj, the **Founder and CEO of Webs Jyoti** - Development & Training Organization is known for his best training and development skills.

He is an innovator in the field of IT and redefined the teaching techniques with a new & traditional approach. In his career of 10 years, he has delivered 350+ websites and taught over 5000+ students.

Hirdesh Bhardwaj developed an interest in writing while he was a fresher in high school. He wrote a Hindi poem "21st Century", later published in June 2009.

As a corporate trainer Hirdesh has conducted more than 50 corporate and seminars so far.

Besides books, Hirdesh has about 250 scripts and tutorials that will be publishing in the upcoming years.

Our Services @ Webs Jyoti

Webs Jyoti is a professional Web development and Information Technology (IT) organization based in Gurgaon, offering web designing and Information Technology services for the global entrepreneurs.

We provide web site development services to clients across INDIA and other part s of the world. We specialize in web site design, Education Software's, e-commerce and CMS.

Thank You

I would like to thank all of these people who has always being supportive from time to time:

- My Parents, my God and Goddess, for their kind wishes and blessings.
- My beloved wife Chanchal, appreciate the support provided by her.
- My 2-year old son Vansh, who thinks that his Dad writes such long books, works for the websites, takes training and seminars, but wishes it included chapters like Cartoon Corner and Ninja's Stories.
- My Sisters Mona and Preeti (Like a daughter too), who wished his brother, would devote more time with them instead of working on this book.
- My Younger Brother Jaiprakash, who always complained that his brother always scold him to work hard and to walk in the early morning to lose his weight.
- My small team @ Webs Jyoti, like a family they think that sir treat them still as student and force to learn something new every day.
- My creative team, P.SerignNdiaye, Jeff Sidabutar andHernan Apraez Pastrana who always ask me to put more and more real time examples from our development house.

Content List

Introduction

Completing this book is one of the hardest things that I have ever done. It took me almost three years to get it into the form you are reading. Every time I read the earlier draft I had to almost always rewrite the whole script. At most places I have tried to show how things work in real time scenarios.

If you ask me to name the most important characteristic of this book I would say simplicity. Be it the text or code, I have tried to make it as simple as I could.

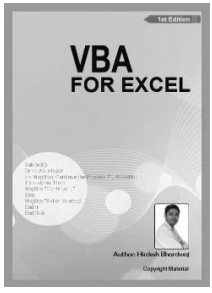
Most of the examples of this book are taken from my training sessions or from our in-house development work.

More than anything else I have tried to design this book for a programmer struggling with a new and complex programming. I have poured my best efforts into these pages. I trust you would find the book useful.

Who Should Read This Book?

This book assumes that you already know web designing or UI Designing using HTML, CSS. If you are completely new to the PHP then I recommend that you read my introductory book on Web Designing-1st Edition Published in August 2016.

Other Books Authored by Hirdesh Bhardwaj



PHP - Introduction

PHP is one of the most popular server -side scripting languages in existence today. If you have a hosting account to run your blog or website, I can guarantee that PHP is already installed on the same server.

PHP code may be embedded into HTML, or it can be used in combination with various web theme systems, web content management system (CMS) and frameworks like CakePHP, Zend, Laravel and Symfony. PHP code is usually processed by a PHP interpreter powered by the Zend Engine.

PHP Facts

- PHP originally stood for Personal Home Page, which is now officially known as 'Hypertext Preprocessor' was released in the year 1995 by Rasmus Lerdorf.
- PHP is free software, Open Source released under the PHP License.
- PHP can be used in all major operating systems including Linux, Microsoft Windows and Mac OS X.
- PHP was originally designed to create dynamic and more interactive web pages. It is the most widely -used, open-source and general-purpose scripting language.
- PHP is installed on over 20 million websites and 1 million web servers.
- 75% of Web 2.0 sites are built in PHP. PHP is used by 81.7% of all the websites whose server -side programming language we know.
- Some of the biggest online brands, such as Facebook, Digg, Flickr, Technorati, and Yahoo! are developed by PHP.

PHP Development Tools

We need the following major tools necessary to develop PHP based application -

Apache Server- A web server that delivers content using HTTP or HTTPS over the World Wide Web (WWW).

MySQL / Mariadb Database - This is a place where you will store your information.

PHP – is a widely used language to produce or developed dynamic web pages.

Various software bundles are there in PHP to fulfill above requirements-

LAMP - Linux, Apache, MySQL, and PHP

WAMP - Window, Apache, MySQL, and PHP

MAMP / **DAMP** - macOS / Mac OS X, Apache, MySQL, & PHP

XAMPP - Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P)

From the above list of bundles I recommend to download and Install **XAMPP**

To work with XAMPP bundle, following steps are required-

-Download & Install XAMPP Latest version.

-Run XAMPP Control Panel, and start running Apache and MySQL Services.

-To test installation of your Apache server, open your Web browser and type http://localhost/

- If your installation was successful, you will see an Apache "success" page in your browser.

PHP IDE / Code Editor

There are various code editors and IDE to work with PHP Environment.

You may use Notepad++, Dreamweaver, NetBeans or sublime text as PHP Code editor.

Customizing Your Installation

The main configuration file for PHP is php.ini; it can be found in the root directory of XAMPP like c:\xampp\apache\php.ini

Some major settings available in php.ini-

short_open_tag: Allows short tags to be parsed (<? and ?>

output_buffering: Allows header lines to be sent after HTML has already been sent to the server. The default is "Off,"

max_execution_time: Sets the limit for how long a script can take to run or execute

max_input_time Sets the limit for how long a script can take to parse the data.

memory_limit Sets the limit for how much memory a script can use to run in MB.

file_uploads: Enables Web site visitors to upload files to your server.

upload_max_filesize: Sets the limit for how large an uploaded file may be, in MB.

mysql.allow_persistent: Determines whether or not a persistent connection can be established with the MySQL server.

mysql.max_persistent: Sets the limit of how many persistent connections are allowed.

PHP Syntax:

PHP programs are written using a text editor, such as Notepad or Dreamweaver, just like HTML pages. We use .php extension to make php file. This extension tells to the server that it needs to parse the PHP code before sending the HTML Response to the Web browser as PHP code are invisible to the people while visiting on your site.

The Rules of PHP Syntax

You should always keep in mind these rules while working with PHP:

- PHP is denoted in the page with opening and closing tags, as follows: <?php and ?>

- PHP lines end or terminates with a semicolon

You can also make program easy by adding comments in your program using double slashes (//) or /* and */

Variables in PHP

Variables are used to store data and can be called by using the variable name and t he data that it holds will be outputted instead. So the variable is something to store information, you use frequently in your program.

The name you decide for the variable should be meaningful. However there are some certain naming rules you need to foll ow while naming for the variable.

Naming Conventions

- Variable name start with the dollar sign
- Must contain only letters, numbers, and the underscore sign
- Cannot start with a numeric value
- Variable name are case sensitive

To minimize confusion and errors , it's best to stick to one naming scheme. The two most common styles are

Use all lowercase, separating words with underscores: $first_name, $emp_code, etc.

Use primarily lowercase, separating words with capital letters: $textName, $fName, etc.

Assigning Value to a variable

Variables can be assigned using the assignment operator (=). Some examples of how to assign value to a variable are below-

```
$empCode = 231;
```

PHP Data Types: Data Types defines the type of data used by variable. PHP allows several types of data types.

Integer: It holds whole numbers including positive and negative numbers Example $rating = 8;

Double: Can hold numbers containing fractional or decimal part including positive and negative numbers. Example $amount=50.85;

String: Hold letters or any alphabets and also numbers Example Name ="Alex";

Boolean: Hold only two values, either TRUE or FALSE

Arrays: Can hold multiple values of same data type Example $arr = array(5,8,12,15);

Objects: Objects are defined as instances of user defined class es that can hold both values and functions.

You can join together string, and whatever is in your variable. The full stop (period or dot, to some) is used for this.

Suppose you want to print out the following "Variable contains the value - 10". In PHP, you can do it like this:

```
<?php

$num = 10;

$text = “Variable contains the value  “;

print($text . $num);

?>
```

So now we have two variables. The $text holds our direct text and $num contain a numeric value i.e. 10. When we're printing the contents of both varia bles, a dot operator is used to separate or concatenate two or more variables.

You can also do this sort of thing:

```
<?php

$num = 10;

echo (“Variable contains the value - “ . $num);

?>
```

This time, the direct text (not inside a variable), but just included in the echo statement. Again a dot (.) is used to separate the direct text from the variable name.

You can also remove information or reset a variable value by assigning null or blank as follow

```
$name = " ";
```

Here, the variable $name exists but does not cont ain a value. It does not mean that $name is set to 0 because 0 is a value. It means that $name does not store any information.

You can go even further and un-create the variable or remove variable from the memory storage by using unset statement:

```
unset($name);
```

After unset statement is executed, the variable $name will no longer exists.

A variable keeps its information for the entire page and not just for a single PHP line. So if a variable is set to “OK” at the beginning of a file, it will still hold “OK” at the end of the page.

Example - To add up using PHP variables

```
<?php
// program to calculate sum of two numbers

$num1 =20;

$num2=30;

$sum=$num1+$num2;

echo “Sum is ”.$sum;
?>
```

Subtraction or multiplication is more or less the same. Instead of the plus sign (+) simply use the minus sign (-) and (*) Change your $sum line –

$sum = $ num1 - $ num2;

Difference between echo and print statement:

- echo has no return value while print has a return value of 1 so it can be used in expressions.
- echo can take multiple p arameters while print can take one parameter.
- echo is faster than print statement.

PHP Superglobals

We already learned about variables, but there are some Ready made or predefined keywords in PHP that can be used without you having to create them first.

Predefined or Superglobals variables are used to provide information from and about the web server, the web browser, and the user.

There are several types of Super Global Variables:-

$_GET - Is used to pass data through URL. Example: http://websjyoti.com?id=2

$_POST - Is used to pass data to server by hiding from URL.

$_REQUEST – Is the combination of $_GET, $_POST, and $_COOKIE.

$_SESSION - is used to maintain session and data from Local machine to Server.

$_COOKIE – Is used to save/get user data on sam e machine in browser cookies.

$_SERVER – Is an associative array which is used to know server information i.e. IP address, Host name, query string, file name etc.

$_FILES – Is used to handle uploaded files operations in PHP. i.e. file size, type, name etc.

Example - $_SERVER Superglobals

```
<?php
  echo "Current IP Address is : ".$_SERVER['REMOTE_ADDR'];

  echo "You are Page at : " . $_SERVER['PHP_SELF'];
?>
```

Please note $_SERVER['REMOTE_ADDR'] in PHP returns current IP address of your system. In case if you a re running this code on your localhost then it should return 1

Functions in PHP

Functions in PHP behave similarly to functions in C or in other programing languages. When we define the functions, we must specify what values the function can expect to receive.

So let's create a function. We need to give the function a name and tell it what variables to expect. We also need to define the function before we call it.

When you use any function it provide the complete reference like how many inputs or values you required to use that function. Let say if I am going to create and use a function it required two inputs i.e.

```
<?php

function sum($a, $b)

{

$output=$a+$b;

Echo “Sum is “.$output;

}

sum(10,5);    // Calling a PHP function

?>
```

So the above sum Function takes two values, in technical term

We called it argument and the function returns the sum of

Numbers, I assigned. Therefore, from the above function we learnt that:

-Function Name is sum

-Function Takes Two Values or argument

-Function Returns output in Numeric

If you are planning to create such functions that takes several arguments and return some of fixed formatted output, then the PHP functions could be the best option.

Apart from user created functions, there are some predefined or in-build functions in PHP that we will learn in next tutorial.

Notes

PHP -Building Blocks

If –else statements

if…else statement – executes set of codes if a condition is true and another code if the condition is false

Within a statement, if a certain condition is true, then something happens. Otherwise, something else happens. We can continue adding else statements as many times as we need.

So in short, "*The IF Else statement checks if the specified condition is being met or not.*"

Syntax

```
if (condition)
  code to be executed if condition is true;

elseif (condition)

  code to be executed if condition is true;

else

  code to be executed if condition is false;
```

Conditional Operators: Sometime also called comparison operator used to compare and test logics and expression.

Following are the conditional operators:

= Equal to

> More than

< Less Than

>= More than and equal

<= Less than and equal

!= or <> Not Equal to

Logical Operators: Logical operators are used to combine multiple statements and to test if the condition met to true / false.

Following are the logical operators:

And or && - All conditions must be true

Or || - at least one should be true

Xor- One side or other must be true but not both

== comparison operator

*Please note = is assignment operator while == is comparison operator in PHP.

Basic If else Example 1

```
<?php
$avg = 50
if ($avg>60)
{
echo "You have a good score";
}
 else
{
echo "Score is less than average";
}
?>
```

Above program will return “Score is less than average” since the value of variable $avg is 50.

Example 2:

```
<?php
$status = 1;
if ($status== 1)
{
print ("<img src =images/right.jpg>");
}
else
 {
print ("<img src =images/cross.jpg>");
}
?>
```

The variable called $status has been assigned a value of 1.The first line of if statement tests to see wh at is inside of the variable called $status. It‘s testing to see whether this variable has a value of 1.

Example 3:

```
<?php
// Below line returns month (1-12) from current date
$month=date(“n”);
if ($month==1 || $month==3 || $month==5 || $month==7 || $month==8 || $month==10 || $month==12 )
```

```
{
echo “31 Days”;
}
else if ($month==2)
{
echo “28 or 29 Days”;
}
else
{
echo “30 Days”;
}
?>
```

We get the value for variable $month by tapping in to built-in date functions. date(“n”) returns a numeric value (1 to 12) of your current month as set in your server, such as 1 for January, 2 for February, and so on.

Loops in PHP

Loops are by far the most powerful component of any programing language. A loop is simply a set of statements repeated more than once.

There may be a situation when you need to execute a block of code several number of times.

But when you are running the same code over and over would produce the same result. So what is the benefit of using Loop in that case?

Basically a loop contains a variable that changes slightly each time.

In PHP several types of Loops are-

-For Loop

-For Each Loop

-Do while Loop

-While Loop

The For Loop

PHP for loop can be used to traverse set of code for the specified number of times. For example if you want to repeat something ten or twenty times.

It should be used if number of iteration or repetition is known otherwise I recommend you to use while loop.

For Loop Syntax

```
for (Initialization; Condition; Progressive)
{
  //set of statements to be executed;
}
```

Parameters Explanation:

Initialization: Initialize the loop counter value

Condition: Check for condition. If it evaluates to TRUE, the loop continues. If FALSE then the loop ends.

Progressive: Increase / Decrease the loop counter value

For Loop Example 1:

```
<?php
for($var=1; $var<=20; $var=$var+1)
{
echo $var ." ";
}
?>
```

Output:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

From the above example $var=$var+1 statement could be simplified something like $var++

Example 2: Display 1st 20 natural numbers in drop-down list

```
<?php
echo "<select>";
for($i=1; $i<=20; $i++)
{
echo "<option>".$i. "</option>";
}
echo "</select>";
?>
```

Please note inside loop area we are using variable $i to display values in drop-down list (Called Select Option field in HTML) as we use variable that changes slightly each time.

The Nested For Loop:

We can use for loop inside another for loop in PHP, it is known as nested for loop. In case of inner or nested loop, nested for loop is executed fully for one outer for loop.

Example, If outer loop i s to be executed for 2 times and nested Inner loop for 2 times, then the nested inner loop will be executed 4 times (2 times for 1st outer loop and 2 times for 2nd outer loop).

So below is the example of PHP nested for loop to design output in a triangle shape

For Loop Example 3:

```
<?php
for($i=1; $i<=5; $i++)
{
for($j=1; $j<=$i; $j++)
{
echo $j. " ";
}
echo "<br>";
}
?>
```

Output

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

Reverse For Loop:

In PHP you can also use For Loop from a higher value to a lower value. To achieve this we need to mention our progressive in negative.

```
<?php
for($var=10; $var>=1; $var=$var-1)
{
echo $var ." ";
}  ?>
```

Output: 10 9 8 7 6 5 4 3 2 1

The While loop

A While loop is used when we have to repeat a set of statements as long as the condition is false. It could be the best where number of repetition is not known earlier or in advance.

It can be explain in plain English as " *Keep doing something until a condition is true".*

Syntax

```
while (condition)
{
//  Set of statements goes here;
}
```

While Loop Example 1

```
<?php
$ct=5;
while($ct<=50)
{
echo  $ct . "<br>";
$ct = $ct+5;
}
?>
```

Output:

5 10 15 20 25 30 35 40 45 50

The do…While Loop in PHP

The do..while loop is almost similar to while loop but one major difference between them. The do...while loop will always execute the block of code at least once.

Difference Between while and do…while Loop: With a while loop, the condition to be evaluated is tested at the beginning of each loop statement, so if the conditional expression evaluates to false, the loop will never be executed.

With a do -while loop, on the other hand, the loop will always be executed once, even if the conditional expression is false, because the condition is evaluated at the end of the loop statement.

The do…while loop example:

```
<?php
$i = 1;
do
{
   $i++;
   echo  $i . "<br>";
}
while($i <= 3);
?>
```

Output: 2 3 4

The foreach Loop in PHP

The foreach loop in PHP widely used with arrays to display elements in the list. The foreach loop runs for the all elements of an array.

Syntax

```
foreach ($array_object as $value)
{
    Set of lines here;
}
```

The Foreach Loop Example

```
<html>
<body>
<?php
$tools=array("WP","OpenCart","Xampp","Laravel", "Zend");
foreach ($tools as $x)
{
    echo $x . " ";
}
echo "Total elements: ".count($tools);
?>
</body>
</html>
```

Output :

WP OpenCart Xampp Laravel Zend

Total elements: 5

Switch Case Statement in PHP

The switch statement is similar to a series of if statements on the same expression.

Switch Case checks a variable for different cases or values. If anyone of the case becomes true then the respective case is executed and the program ignores all other cases.

For example when you have call to a customer care, in the very beginning you have some recorded messages like “for billing support press 1, for technic al support press 2, to go back to main menu press 9 and so on...” how this concept work ?

It is just an example of Switch case where each key is bind with some selective operations.

Switch Case Example 1

```
<?php

$i=0;

switch ($i)

{

case 0:

echo "$i equals 0"."<br/>";

break;

case 1:

echo "$i equals 1"."<br/>";

break;
```

```
case 2:
echo "$i equals 2"."<br/>";
break;
}
?>
```

Switch Case Example 2

```
<html>
<head>
<title>Switch Case Example in PHP</title>
</head>
<body>
<?php
$f=10;
$s=5;
$choice='-';   // you can change value of $choice + / - / *
switch ($choice)
{
case "+":
$sum=$f+$s;
echo "Sum=".$sum;
break;
```

```
case "-":
$sub=$f-$s;
echo "Subtraction=".$sub;
break;
case "*":
$mult=$f*$s;
echo "Multiplication=".$mult;
break;
case "/":
$div=$f/$s;
echo "Division=".$div;
break;
default:
echo "Invalid choice";
}
?>
</body>
</html>
```

Notes

Data Sending Ways in PHP

There are some certain ways to send you r information or data to the PHP page-

1. Using Form Methods - Get & Post

2. Using Cookies & Session

3. Using Query String / URL Rewriting

4. Using Hidden form field

Form Methods

The GET Method: The GET method sends information separated by the ? Operator.

Example: http://www.xyz.com/home.html?city=Gurgaon

Please note that t he GET method is restricted to send upto 1024 characters (1024 bytes) only. You should not use GET method if you have password or other confidential information to be proceeding over the server.

GET can't be used to send binary data, like im ages or word documents, to the server.

The PHP provides $_GET associative array or super global variable to access the data sent using GET method.

Example –GET Method:

Let‘s create a HTML form that contain a form element-Text Field

HTML Page: form.html

<form action =”thanks.php” method=”GET”>

Enter Name : <input type=”text” name=”txtName”>

<input type=”submit” value=”Submit”>

</form>

Explanation: Here we have a text field called txtName, basically we wanted to send the value of this field to the PHP page called thanks.php. Make sure your form should contain a button type submit to send information to action page.

Let‘s create a page thanks.php in the same directory where we have the above html page

PHP Page: thanks**.**php

<?php

echo $_GET[‘txtName’];

?>

Meanwhile you can see the same entered value on the browser‘s address bar. Here we have print the same value received through text field named txtName. Make sure the same name to be put inside [ ] of super global variable $_GET as mentioned in HTML Form.

The POST Method

The POST method transfers information via HTTP headers.

The POST method does not have any restriction on data size to be sent. So it can be used to send files, images as well as binary data as well.

The PHP provides $_POST super global va riable or associative array to access data sent using POST method.

Example –POST Method:

Let‘s create a HTML form that contain several form elements

HTML Page: form.html

<form action =”thanks.php” method=”POST”>

Enter Email : <input type=”email” name=”email”>

Enter Password : <input type=”password” name=”pass”>

<input type=”submit” value=”Login”>

</form>

Here we have two fields email field called email, password field called pass, basically we wanted to send the se above values to the PHP page called thanks.php

Let‘s create a page thanks.php in the same directory where we have the above html page

PHP Page: thanks**.**php

```
<?php

echo $_POST[‘email];

echo $_POST[‘pass];

?>
```

Here we have print the same value received through email and password field respectively.

We can make it little advanced by taking above posted values to the variables. The advantage of using variable will be like we can use some validation and sanitizing data before preceding it to the database.

So the above code can be modified something like this-

```
<?php

// collect values in the variables separately

$email= $_POST[‘email];

$password= $_POST[‘pass];

// you may use some validations like

if($email==”” || $password==””)

echo “Input required”;

else

echo “Welcome “.$email . “ Your password is “.$password;

?>
```

The $_REQUEST variable

The PHP $_REQUEST variable contains the contents of both $_GET, $_POST, and $_COOKIE. We will discuss $_COOKIE variable when we will explain about cookies.

The PHP $_REQUEST variable can be used to get the result fro m form data sent with both the GET and POST methods.

Example

```
<?php
  if( isset($_REQUEST["name"]) )
 {
    echo "Welcome ". $_REQUEST['name']. "<br />";
    echo "You are ". $_REQUEST['age']. " years old.";
  }
?>

<html>
 <body>
    <form action = “ " method = "POST">
       Name: <input type = "text" name = "name" />
       Age: <input type = "text" name = "age" />
       <input type = "submit" />
    </form>
     </body>
```

```
</html>
```

Here we can also use $_PHP_SELF variable in form action that contains the name of self script in which it is being called.

Using Query String

The query string can be passed from one page to another by appending it to the URL. You can pass more than one query string by separating the & sign between the query strings.

A query string can contain two things: the query string ID and its value. If you want to access the query string you can use $_REQUEST variables.

Example: test.html

```
<html>
<head>
<title>Query String Example</title>
</head>
<body>
<a href="view.php?city=Gurgaon&state=Haryana">View
Location</a>
</body>
</html>
```

Let‘s create a PHP page view.php which should receive two variables values as follow-
city =Gurgaon, state=Haryana

view.php

```
<?php

echo "Location is ".$_REQUEST['city'] . " " .$_REQUEST['state'];

?>
```

You can also retrieve everything after the ? And assign it to a variable, we can simply use $_SERVER super global like so:

```
<?php

$txt = $_SERVER['QUERY_STRING'];

var_dump($txt);

?>
```

Please note, if there is no query string, then the QUERY_STRING key in $_SERVER will be an empty string

Notes

PHP - Strings Functions

String functions are the most important functions frequently used in PHP. String is the sequence of characters, like "Let‘s learn String functions”. There is no installation required to use string function.

Example

```
<?php
  $name = "Hirdesh Bhardwaj";
  print($name);
 ?>
```

String Concatenation Operator

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

```
<?php
  $txt1="Hello Readers";
  $txt2="8756";
 echo $txt1 . " " . $txt2;
?>
```

Using the strlen() function

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

```
<?php
$txt=”Webs Jyoti”;
```

```
  echo strlen($txt);
?>
```

This will produce the following result – 10 (including space)

Using the strpos() function

The strpos() function is used to search for a text or character within a string. If the value is found then this will return the position of the first value. If no match is found, it will return FALSE.

Let's see if we can find the string "PHP" in our string −

```
<?php
  echo strpos("Hello PHP","PHP");
?>
```

This will produce the following result – 6

The explode() Function: Used to convert strings into an array variable. It can be used in case where the information that you want lies in a larger string, you can extract it out with the explode() function.

Example:

```
$db = explode(';',"host=localhost;db=hirdesh;uid=root ");
print_r($settings);
```

Output:

Array ([0] => host=localhost [1] => db=hirdesh [2] => uid=root [3]

Another Example:

```
<?php

$text = "Lets Learn PHP Strings";

echo (explode(' ',$text));

?>
```

Output

```
Array
(
[0] => Lets
[1] => Learn
[2] => PHP
[3] => Strings
```

The substr() Function:

The substr() is used to return part of the string. It accepts three (3) basic parameters. The first one is the string to be shortened, the second parameter is the position of the starting point, and the third parameter is the number of characters to be returned.

Example:

```
echo str_replace(' ', '-', 'PHP Tutorial');
```

It will replace Space by - and the final output will be like

PHP-Tutorial

The md5( ) Function: Used to calculate the md5 hash of a string value. This could be used where you want to convert your password from plain text to hash code to prevent the use of unauthorized users.

Example

```
<?php

echo md5('password123');

?>
```

Output: 9f161034ee4de758baf4de09c4eb1a75

The trim() function : the Trim() is used to remove white spaces and predefined characters from a both the sides of a string.

Syntax

```
trim(string,charlist)
```

Example

```
<?php

$txt = “Webs Jyoti”;

echo trim($txt,”Weti”);

?>
```

Output

```
bs Jyo
```

The str_replace() function: This can be used where you just want to swap out words or a set of characters in a string and replace them with something else.

Example:

```
<?php

$oldstr = "The cat is black";

$newstr = str_replace("black", "white", $oldstr);

echo $newstr;

?>
```

Output: The cat is white

Example: String Functions- All in one

Task: Collect Name and Contact no. from user and proceed for the validations

form.html

```
<form action="viewdata.php">
  Enter Name : <input type="text" name="txtName" />
  <br />
  Phone : <input type="text" name="phone" />
  <br />
  <button type="submit">Send Value</button>
</form>
```

viewdata.php

```
<?php
$name= $_REQUEST['txtName'];
$p=$_REQUEST['phone'];
$l=strlen($name);
if(strlen($p)<>10)
{
echo "Invalid mobile number ";
//exit;
}
 echo "Length of your Name is ".$l;
 echo "<br>First Name is ".substr($name,0,strpos($name," "));
 $lft=substr($p,0,2);
 $last=substr($p,8,2);
  echo "<br> OTP should send to ".$lft."XXXXXX".$last;
?>
```

Notes

PHP - Date / Time Functions

When working in any programming language, d ealing with dates and time is a common task, we frequently used in PHP.

PHP's date () function gives you all the information that you need about the current date.

```
<?php
$today = date("d/m/Y");
echo "Today is ".$today;
?>
```

We can use different time format with separator like d-m-Y, d.m.Y and so on.

The DateTime class in PHP : The DateTime class in available in PHP 5 and higher versions.

```
<?php
$d  = new DateTime;
echo $d->format('Y-m-d H:i:s');
?>
```

The above example will display today's date and time in yyyy-mm-dd h:m:s format. If you are getting warning saying not to rely on system time zone, then add following line to the code.

date_default_timezone_set(' Asia/Kolkata ');

Example: To automatically refresh the copyright year on your website- © -<?php echo date("Y");?> All rights reserved.

The PHP time() :_ The time() function is used to get the current time as a Unix timestamp

Example:

```
<?php
$cDate = 1394003958;
echo(date("F d, Y h:i:s", $cDate));
?>
```

Example: To print date like 10 July 2015

```
<?php
$date1 = date_create("2016-05-10");
 echo date_format($date1, 'j-F-Y');
?>
```

Output: 10 May 2016

Code Examples	**Output**
$d = new DateTime('now'); echo $d->format('Y-m-d') ;	Current Date 2010-12-20
$d = new DateTime('tomorrow'); echo $d->format('Y-m-d') ;	Tomorrow‘s Date
$d = new DateTime('yesterday'); echo $d->format('Y-m-d') ;	Yesterday‘s Date
$d = new DateTime('+1 day'); echo $d->format('Y-m-d') ;	Next date from current system date (Try following code +1 month, Monday +1 day, +2 years 10 days, +3 years

PHP Timezones

If the PHP DateTime is not matching with your current time then It might happen if you are using a server that is located in a different timezone.

The **date_default_timezone_get()** is used to get the default timezone, used by all date/time functions in a script.

To get default or the current time zone in PHP use below code

```
<?php
 $tz = date_default_timezone_get();
   echo "Default timezone: ".$tz;
?>
```

To setting the timezone use below code

```
<?php
  date_default_timezone_set("America/Los_Angeles");
  echo "The Current time is " . date("h:i:sa");
?>
```

Notes

Arrays in PHP

An array is a data structure that stores si milar type of values in a common variable. Arrays are also called homogeneous data type.

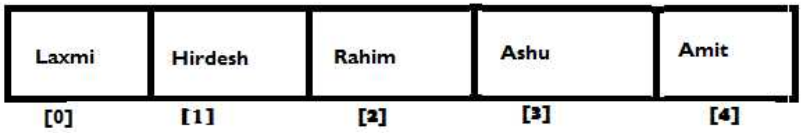
For example if you want to store 50 numbers then instead of defining variables it is easy to define an array with size of 50.

In PHP, the array() function is used to create an array

Arrays can store numbers, strings and any object but their inde x will be represented by numbers. By default array index starts from zero.

Example

$x=array("Laxmi","Hirdesh","Rahim","Ashu","Amit");



From the above diagram, values showing in each boxes called index in array. Insertion of values in array always starts f rom first index i.e. 0.

To count numb er of elements stored in array count() method is used in PHP Arrays

echo count($x);

After the array is declared and assigned, you can populate value from any specific index as follow

echo $x[1]; // output – Hirdesh

To print all elements present in array x you can use for-each or simply for loop as follow

Using foreach loop

```
foreach ($x as $val)

{

echo  $val . " ";

}
```

Using For loop

```
for($i=0; $i<count($x); $i++)

{

echo $x[$i] . "<br>";

}
```

Sorting Arrays: PHP provides many easy ways to sort array values in predefined orders (Ascending and Descending)

arsort: Sorts the array in descending value order and maintains the key/value relationship

asort: Sorts the array in ascending value order and maintains the key/value relationship

rsort: Sorts the array in descending value order

sort: Sorts the array in ascending value order

Examples - Sorting of arrays

```
$x=array("laxmi","hirdesh","rahim","ashu","amit", "mohit");

// rsort for DESC and simply sort for ASC

sort($x);

foreach ($x as $val)

{

echo  $val . " ";

}
```

Associative Arrays

The associative arrays are similar to default arrays in term of functionality but they are different in terms of their index. Associative array will have their index as str ing so that you can bind them between key and respective values.

asort for displaying in ASC by value (numeric value) the same arsort for DESC

```
echo "<br> ASort <br>";

$emp = array("hirdesh"=>"12", "Amit"=>"54", "Laxmi"=>"50", "Rahim"=>"87");

asort($emp);

foreach($emp as $x => $code)

 {
```

```
echo "Name=" . $x . ", Emp Code=" . $code ." <br>";

}
```

ksort for displaying in ASC by Key (names or first key string) the same krsort for DESC

```
echo "<br> KSort <br>";

$emp = array("Hirdesh"=>"12", "Amit"=>"54", "Laxmi"=>"50", "Rahim"=>"87");

ksort($emp);

foreach($emp as $x => $code)

 {

    echo "Name=" . $x . ", Emp Code=" . $code ." <br>";

}
```

Notes:

Sanitize and Validations Functions

Data validation is the major part of working with forms. Not only can invalid submitted data lead to security problems, but it can also break your page and output. So before proceeding or inserting data into database you need to validate it to avoid any error.

filter_var function in PHP

filter_var used for both, sanitize and validate data. Sanitizing will remove any illegal character from the data and Validating will determine if the data is as per expected or in proper form.

How to use filter_var

Using filter_var is incredibly easy. It's simply a PHP function that takes two inputs of data:

The variable you want to check and The type of check to use

Example

```
$title = "<h1>Lets learn PHP Functions <br></h1>";
$ftitle = filter_var($title, FILTER_SANITIZE_STRING);

echo "After SANITIZE data is ".$ ftitle;
```

Output : Lets learn PHP Functions

Example to check for valid IP Address

```
$ip = "127.0.0.1.33";

$valid_ip = filter_var($ip, FILTER_VALIDATE_IP);

if($valid_ip)

echo "valid IP Address ";

else

echo "invalid IP Address ";

// to check url received in proper format ?

//$homepage = filter_var($_POST['homepage'],

FILTER_SANITIZE_URL);
```

Example to check email received in proper format

```
echo "<br>";

$x="info@websjyoti.om";

$email = filter_var($x, FILTER_SANITIZE_EMAIL);

if (filter_var($email, FILTER_VALIDATE_EMAIL))

{

echo "email id is a valid .<br/><br/>";

}

 else
 {
echo "$x is NOT a valid email address.<br/><br/>";
  }
```

Example to check given value is integer

```
$val = 125;

// validate the integer

echo filter_var($val, FILTER_VALIDATE_INT);

// in case if you put string value then this will return nothing because value is false. Because variable has failed validation
```

Example form Validation (All in One)

```
<?php
   if (isset($_POST['email'])) {
      echo filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
      echo "<br/><br/>";
   }
   if (isset($_POST['homepage'])) {
      echo filter_var($_POST['homepage'], FILTER_SANITIZE_URL);
      echo "<br/><br/>";
   }
?>
```

```
<form name="form1" method="post" action="form-sanitize.php">

   Email Address: <br/>

   <input type="text" name="email" value="<?php echo
$_POST['email']; ?>" size="50"/>

<br/><br/>

   Home Page: <br/>

  <input type="text" name="homepage" value="<?php echo
$_POST['homepage']; ?>" size="50" /> <br/>

   <br/>

   <input type="submit" />

</form>
```

In order to ensure the data is properly formatted, it needs to be validated

```
<?php

   if (isset($_POST['email'])) {

      $email = filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL);

      if (filter_var($email, FILTER_VALIDATE_EMAIL)) {

         echo "$email is a valid email address.<br/><br/>";

      } else {

         echo "$email is <strong>NOT</strong> a valid email
address.<br/><br/>";
```

```
        }
    }
    if (isset($_POST['homepage'])) {
        $homepage = filter_var($_POST['homepage'],
FILTER_SANITIZE_URL);
        if (filter_var($homepage, FILTER_VALIDATE_URL)) {
            echo "$homepage is a valid URL.<br/><br/>";
        } else {
            echo "$homepage is <strong>NOT</strong> a valid
URL.<br/><br/>";
        }
    }
?>
<form name="form1" method="post" action="form-validate.php">
Email Address: <br/>
<input type="text" name="email" value="<?php echo
$_POST['email']; ?>" size="50"/> <br/><br/>
<input type="text" name="homepage" value="<?php echo
$_POST['homepage']; ?>" size="50" /> <br/>
<input type="submit" />
/form>
```

Notes :

Introduction to Mysql / Mariadb

PHP MariaDB formally known as PHP MYSQL developed in 2009 is high compatible and exact matching with MySQL APIs and commands. The original developers of MySQL created MariaDB after concerns raised by Oracle's acquisition of MySQL. MariaDB (and its predecessor MySQL) has been used around the world for over 20 years

Introduction to Relational Databases:

A database is as a collection of related files. A database and the tool that controls the database, called a database management system (DBMS). Most databases today are relational databases because they deal with tables containing related fields.
So let‘s understand a few terms related to databases:

Database − A database is a data source containing tables of related data.

Table − A table, like a spreadsheet or excel sheet, containing data in tabular way.

Column − A column, is a structure holding data of one type.

Row − A row, sometime also known as a tuple or record is a structure grouping related data.

Primary Key − this refers to a unique, identifying value. This value cannot repeat within a table, and there can have only one primary key in the table.

Foreign Key − a foreign key serves as a link or interfac e between two tables.

Compound Key − A compound key is a key that refers to multiple columns.

Data types in Mysql / Mariadb :

Numeric Data Types:

INT (Integer): Int data type represents an integer of normal size. So simply it can used to store numeric val ue. There are also BIGINT, MEDIUMINT, SMALLINT and TINYINT which represent different range of integer values.

DECIMAL: Decimal data type represents numbers with specific floating values. Maximum permitted value is 65 and maximum decimals are 30. There are also FLOAT and DOUBLE to store numeric values with floating point.

BOOLEAN: This data type associates a value 0 with “false,” and a value 1 with true.

Date and Time Data Types:

DATE: The DATE data type represents a date in “YYYY-MM-DD” format.

TIME: The TIME data type represents a time in "hh:mm:ss" format.

DATETIME: The DATETIME data type represents date and time in “YYYY-MM-DD HH:MM:SS” format.

TIMESTAMP: This data type represents a timestamp of the “YYYY-MM-DD HH:MM:DD” format. it mainly used to insert current date and time of record insertion / updatation and deletion process.

String Data Types

CHAR: This data type represents a fixed-length string. default value is 1.

VARCHAR: This data type represents a variable-length string of 0 to 65535.

TEXT: This data type represents a text column with a maximum length of 65,535 characters. some more Text data types are MEDIUMTEXT, LONGTEXT and TINYTEXT.

ENUM: The ENUM data type represents a string object and allows only a single value from a given list.

SET: The SET data type represents a string object having zero or more values from a given list.

Types of Mysql/ MariadbTables

Now that you understand some of the general features of tables, you should know that there are different types of tables.

There are some main types of tables in the current version of Mysql:

MyISAM-A default table and will usually be sufficient for the user‘s needs. It supports all the field types and the parameters.

ISAM- It is same as the MyISAM table, except that it can‘t handle data larger than 4GB this means it isn‘t portable across operating systems. This table type is no longer be available in PHP5.

HEAP - These are mostly used for temporary tables they don‘t support a common features of the MyISAM table, such as auto_increment and blob/text columns etc.

This can be used when your are working with log data and don't want to store it for a long time.

InnoDB- is considered to be “transaction safe,” which means that you can recover data from crashes.

Some popular MySQL commands

CREATE: Creates new databases and tables

ALTER: Modifies existing tables

SELECT: Display the data you want

DELETE: Erases or Delete the data from your table

INSERT INTO...: Puts values into the table

UPDATE: Lets you modify data already in a table

DROP: Deletes an entire table or database

Working with MYSQL / Maraidb Database

Before implementing Mysql with PHP lets explore the basic concepts of database and table through MYSQL Console.
-Open XAMPP Control Panel and start Apache and MYSQL\Mariadb services

-Open Command prompt (type cmd on run option) and copy paste your MYSQL/Mariadb path using cd command like
cd C:\xampp\mysql\bin

-To open MYSQL environment type mysql.exe –u root
(Here root is the username of your Mysql database . If your Mysql contain password then type mysql.exe –u root –p mypassword)
- Once your MYSQL welcome screen is open then you may start working with Mysql / maridb environment.

MYSQL Popular Commands

To open the list of databases present in Mysql Or in Mariadb
show databases;

To Create new database name named "Chanchal"
create database Chanchal;

To use or enter into the newly created database named "Chanchal"

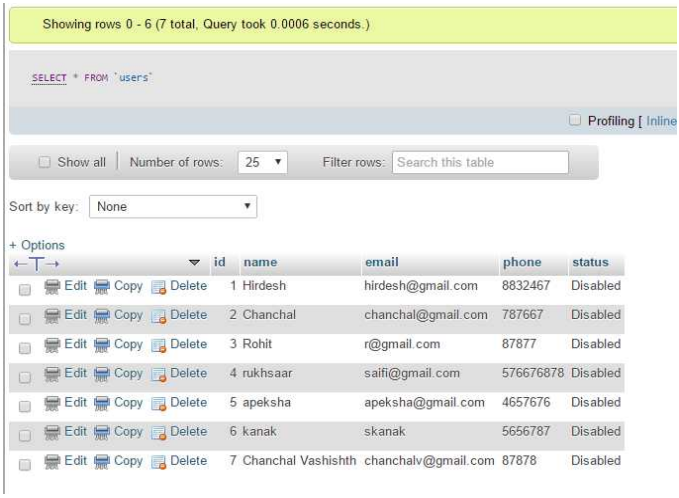
use Chanchal;

To check how many tables are there in the present database?

show tables;

To insert record in table like in columns i.e. name, email and phone

insert into users(name,email,phone) values ('chanchal', 'chanchal@gmail.com', '878787');



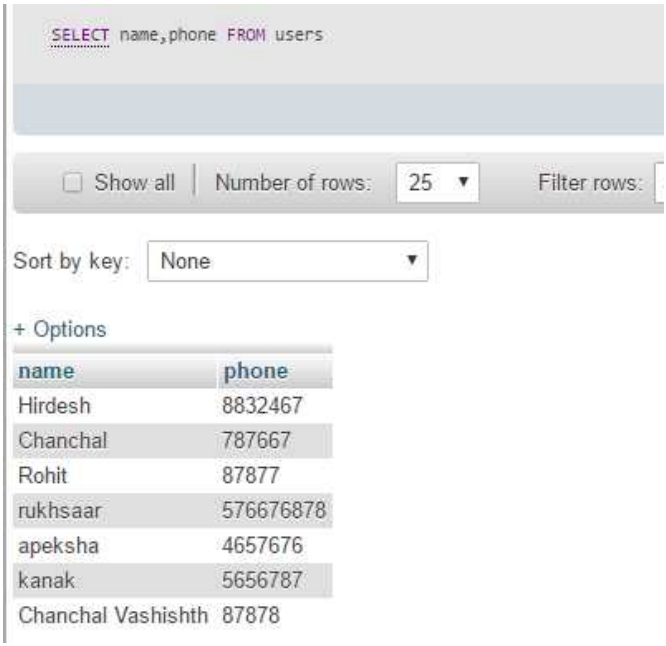
	id	name	email	phone	status
Edit Copy Delete	1	Hirdesh	hirdesh@gmail.com	8832467	Disabled
Edit Copy Delete	2	Chanchal	chanchal@gmail.com	787667	Disabled
Edit Copy Delete	3	Rohit	r@gmail.com	87877	Disabled
Edit Copy Delete	4	rukhsaar	saifi@gmail.com	576676878	Disabled
Edit Copy Delete	5	apeksha	apeksha@gmail.com	4657676	Disabled
Edit Copy Delete	6	kanak	skanak	5656787	Disabled
Edit Copy Delete	7	Chanchal Vashishth	chanchalv@gmail.com	87878	Disabled

To display All records from Table Called - Users

SELECT * FROM users order by name DESC;

To display selective columns from the table Only Name and Phone

SELECT name,phone FROM users;



name	phone
Hirdesh	8832467
Chanchal	787667
Rohit	87877
rukhsaar	576676878
apeksha	4657676
kanak	5656787
Chanchal Vashishth	87878

To display All Records in ascending order by ID column

SELECT * FROM users order by id ASC;

To Filter Records using where clause

SELECT * FROM users WHERE name='chanchal' ; (Exact Match)

OR

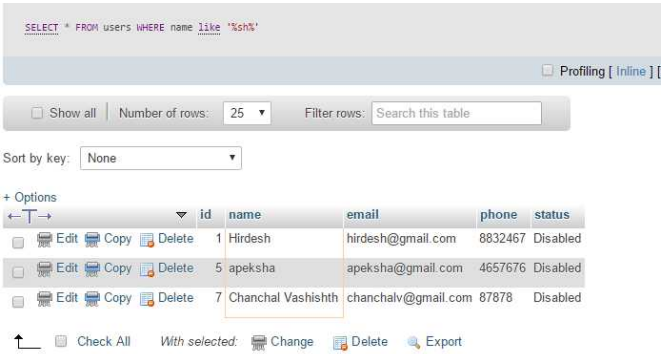
SELECT * FROM users WHERe id >3; (Using comparison operator)

To get the list of columns in the table or you want to see the structure of table

show columns from users;

To find or search record (Approx search)

SELECT * FROM users WHERE name like '%sh%' ;



To update values from table - change the name of a record called chanchal sharma

update users set name='Chanchal Sharma where id=7;

To delete a record from table - Users

delete from users where id=7;

To clear all records or empty a table

truncate users;

To drop or delete database named Hirdesh

drop hirdesh;

Aggregate Functions in MYSQL

Aggregate functions perform a calculation on multiple rows and allow us to easily produce summarized data from our database or table.

Some aggregate functions in Mysql, s uch as SUM, COUNT, MAX, MIN and AVG

SUM Function: SUM function returns the sum of a numeric column. Let‘s calculate the total amount of salary in our table.

Example:

SELECT SUM(salary) as 'Total Salary' from mytable;

COUNT Function: COUNT function return s a count of rows or records in a column. Let‘s count the total no. of employees working in IT department.

Example:

SELECT COUNT(name) from mytable where dept=‘IT‘;

The DISTINCT Keyword: The DISTINCT keyword that allows us to omit duplicates and return unique records from a column. The same can also achieve by grouping **(Group By Clause)** similar values together.

Example:

SELECT DISTINCT (city) FROM mytable;

Using Group By clause

The group by widely used with aggregate functions like sum, avg, max min count etc. it create unique group of each element present in the column.

Example

select dept, sum(salary) from emp_table group by dept;

Above query will display the sum of salary from each department in a pivot view.

Group By clause and Distinct keyword

Group By	Distinct
Group by clause widely used with aggregate functions like sum, avg, max min count etc. it create unique group of each element present in the column	Distinct keyword used in simple query to fetch with where clause to get unique records. Distinct cannot use for calculation or aggregation purpose.

The Having Clause: The HAVING clause can be used to filter records after a GROUP BY clause is applied. So it can be used to summarize group records. So Only the groups(created after group by claus e) that meet the HAVING criteria will be returned as a result.

How the having clause is different from where clause? HAVING applies to summarized group records, whereas WHERE applies to individual records. In short " *where clause never work with aggregate function with group by*".
The where clause filter record before the aggregate of data while the having perform after aggregation/calculations is performed.

The Having Clause syntax

SELECT column-names FROM table-name WHERE condition GROUP BY column-names HAVING condition

Example: List the number of employees in each department.
Only include department with more than 15 employees.

SELECT COUNT(name), dept FROM mytable GROUP BY dept HAVING COUNT(name) > 15

Example 2

select min(salary) from emp group b y emp_name having min(salary)<15000;

Example 3

select dept,count(*) from emp group by dept having count(*)>3;

Subqueries in MYSQL : A Sub query or Inner query sometime called as Nested query, is a query within another SQL query and embedded within the WHERE clause. In Simple words, a Subquery is an inner query which is placed within an outer query using different SQL clauses.

We can use Subquery anywhere mostly with select command in MYSQL enclosed within parentheses. An ORDER BY command cannot be used in a subquery, although the main query can use an ORDER BY. The GROUP BY command can be used to perform the same thing..

Example: Find list of names from a department who have more than 3 employees in the dept.

```
select name from emp where dept in(select dept from emp group by dept having count(*)>3);
```

Example 2: Display highest salary department wise and names who is taking maximum salary.

```
select emp_name from emp where salary in(select dept, max(salary) from emp group by dept);
```

```
SELECT employee_id,first_name,last_name,salary
    FROM employees WHERE salary >
            (SELECT AVG(SALARY) FROM employees);
```

Notes

PHP-Insert Data into Mariadb/Mysql

In the previous tutorial we learned how to send HTML form data to a PHP Page. Now we have to store the same information into MYSQL or Mariadb database.

To complete the above task we have to follow the below steps-

- Design a HTML Page containing HTML Form fields
- Receive form field data to the PHP page using $_GET, $_REQUEST or $_POST
- Prepare a database connection
- Place appropriate query to insert record into Mysql table

So let‘s create a sample HTML page containing essential form fields

add-employee.html

```
<html>
<head>
<title>Add Employee Details</title>
</head>
<body>
<form method="post" action="thanks.php">
<table border="1" width="350">
<tr>
<td>Enter Name</td>
<td><input type="text" name="txtName"></td>
</tr>
```

```
<tr>
<td>Enter Email</td>
<td><input type="email" name="txtEmail"></td>
</tr>
<tr>
<td>Enter Password</td>
<td><input type="text" name="txtPassword"></td>
</tr>
<tr>
<td>Enter Phone</td>
<td><input type="text" name="txtPhone"></td>
</tr>
<tr>
<td colspan="2" align="center">
<input type="submit" value="Add Record">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Now let‘s create an action page called – thanks.php to receive data from the above HTML Form Page. Make sure that you should have a submit button to proceed.

thanks.php

```
<?php
// create a variable
$name=$_REQUEST['txtName'];
$email=$_REQUEST['txtEmail'];
$password=$_REQUEST['txtPassword'];
$phone=$_REQUEST['txtPhone'];
// create a database connection
$con=mysqli_connect("localhost","root"," ","mydb");
// test database connectivity
if(mysqli_connect_errno($con))
{
echo 'Failed to connect...';
}
//Prepare and execute the insert query
$q= mysqli_query($con, "insert into
employees(name,email,password,phone) values
('$name','$email','$password','$phone')");
if($q)
{
echo "Record inserted successfully";
}
else
{
echo mysqli_error($con);
}
mysql_close($con);
?>
```

Now let‘s create a Database and table (Open MYSQL localhost://phpmyadmin) as required below

Database Name: mydb

Table Name: employees

Column List: name, email, password, phone

To create a above database mydb run the following command

Create database mydb;

To create table employees containing columns as mentioned above run the following command -

Create table employees (name varchar(20), email varchar(30), password varchar(20), phone varchar(25));

Connecting to the MySQL Server

Before you can do anything with MySQL, you must first connect to the MySQL server using your specific connection variables. Connection variables consist of the following parameters, which you must know before you make connectivity to the server:

Host name- In our case, it‘s the localhost because we‘ve installed everything locally. You will need to change this to whatever host is acting as your MySQL server.

User name- We‘ve used just the root user for local computer, but you may have another user name that you‘ll need to create username while making site live.

Password- We have just blank " " password for local computer, but you need to use your own password while making site live.

So you may put the same configuration into PHP Page something like this-

```
<?php
$host = “localhost”;
$user = “root”;
$pass = “ ”;
$db="mydb";
$connect = mysqli_connect($host, $user, $pass,$db);
OR
$connect = mysqli_connect(“localhost”, “root”, “”,"mydb");
?>
```

Enter Name	
Enter Email	
Enter Password	
Enter Phone	
Add Record	

Notes

Retrieving Data from Mysql/Mariadb

In the previous lesson, we learnt how to send data to the database. Now it's time to retrieve data from our database to our PHP pages.

This is really one of the most important lessons in this book. Once you have read and understood this lesson, you will realize why database-driven web applications are so robust.

To retrieve data from a database, you use select queries like -

*SELECT * FROM employees*

Data display from database to the web page will process in two steps -

- The very first it will count the number of rows. You should know the number of the records in your table.

- Secondly we have to set the loop accordingly. Each time it will be incremented by one. It will fetch each row of the result and print the data stored there.

There are some popular methods to retrieve records from the database in PHP-

mysql_fetch_row(): It will fetch result row as an numeric way. it returns a row where the values will come in the same order as they are defined in the SQL query.

mysql_fetch_assoc():It will fetch a result row as an associative array. This function will return a row as an associative arr ay where

the column names will be the keys storing corresponding value. Means we can fetch our record through the table column name.

mysql_fetch_array(): It will fetch a result row as an associative array or a numeric array. So this is the combination of b oth the method of mysql_fetch_row and mysql_fetch_assoc.

mysql_fetch_object(): It will fetch a result row as an object

Example 1: To retrieve data from Mysql Table - Lets retireve records of recently added table as below-

Database Name: mydb
Table Name: employees
Column List: name, email, password, phone

```
<?php
// Prepare a database connectivity first
$con = mysqli_connect("localhost", "root", " ", "mydb");
// Prepare SQL Statements to fetch record
$query = mysqli_query("SELECT  * from employees";
// Use any fetch method to retrieve records
  while ($row = mysqli_fetch_assoc($query))
 {
/* Now you can display columns using $row variable as all records to be proceed by the $row variable */
```

```
echo $row[‘name’]. “<br>”;
echo $row[‘email’]. “<br>”;
}
/* close connection */
mysqli_close($con);
?>
```

Example 2: Using mysqli_fetch_array()

```
<?php
//Connect to database
$con = mysqli_connect("localhost", "root", "", "mydb");
//execute simple query
$result = mysqli_query("SELECT * FROM employee",$con);
while ($row = mysqli_fetch_array($result))
 {
//access array with column number index
echo $row[0]. " " .$row[1];
}
mysqli_free_result($result);
?>
```

Example 3: Display records (names) in drop down

```
<?php
//Connect to database
$con = mysqli_connect("localhost", "root", "", "mydb");
//execute simple query
$result = mysqli_query("SELECT * FROM employee",$con);
echo “<select style=’width:150px’>”;
```

```
while ($row = mysqli_fetch_array($result))
{
echo “<option>”.$row[0]. “</option>”;
}
mysqli_free_result($result);
?>
```

Example 4: Display MYSQL records in HTML Table

```
<?php
//Connecting to the database
$con = mysqli_connect("localhost", "root", "", "mydb");
//execute simple query
$result = mysqli_query("SELECT * FROM employee",$con);
?>
<table border=”1” width=”400” cellpadding=”5”>
<?php
while ($row = mysqli_fetch_assoc($result))
 {
?>
<tr>
<td><?php echo $row[‘name’]; ?></td>
<td><?php echo $row[‘email]; ?></td>
<td><?php echo $row[‘phone]; ?></td>
</tr>
<?php
}
?>
```

</table>

Example 4: Basic Searching operation using MYSQL where clause

HTML Form

<html>

<head>

<title>Simple Search</title>

</head>

<body>

<form action=”search-records.php” method=”get”>

Name : <input type=”text” name=”txtName” placeholder=”Enter few letters”>

<input type=”submit” value=”Search”>

</form>

</body>

</html>

Search.php

<?php

$name=$_REQUEST[‘txtname’];

// Connecting to the database

$con = mysqli_connect("localhost", "root", "", "mydb");

//execute simple query

```
$result = mysqli_query("select  * from employee where name like
‘%$name%’ ",$con);

?>

<table border=”1” width=”400” cellpadding=”5”>

<?php

while ($row = mysqli_fetch_assoc($result))

 {

?>

<tr>

<td><?php echo $row[‘name’]; ?></td>

<td><?php echo $row[‘email]; ?></td>

<td><?php echo $row[‘phone]; ?></td>

</tr>

<?php

}

?>
</table>
```

Notes

Error Handling in PHP

Error handling is the process of catching errors from the program and then taking appropriate action. Before proceeding for error handling you should get to know how many types of errors occurred in PHP.

Notices: These are non-critical errors that occurred while executing a script - for example, accessing a variable that has not yet been defined. By default, such errors are not displayed to the user at all - although, as you will see, you can change this default behavior.

Warnings: These are more serious errors - for example, attempting to use a file (using include() method) which does not exist. By default, these errors are displayed to the user, but they do not result in script termination.

Fatal Errors: These are critical errors - for example, creating an object of a non-existent class, or calling a function which doesn't exist in your program. These errors cause the immediate termination of the script, and PHP‟s default behavior is to display them to the user when they take place.

It‘s very simple in PHP to handle errors.

Using die() function

While writing your PHP program you should check all possible error condition before going ahead and take appropriate action when required.

Example

```
<?php
  if(!file_exists("/docs/resume.txt"))
 {
     die("File not found");
  }
else
 {
     $file = fopen("/docs/resume.txt","r");
  }
?>
```

Example**:**

```
<?php
$a = 0;
$b = 200;
$result = $b/$a;
echo "RESULT: ". $result;
?>
```

Output: Warning: Division by zero in ... on line 4

Example:

```
<?php
echo "Hi lets learn PHP"
echo "<br/>This is another line";
?>
```

Output: Parse error: syntax error, unexpected 'echo' (T_ECHO), expecting ',' or ';' in ... on line 3

Some possible errors in PHP

E_ERROR - Fatal Run-time errors. Execution of the script is halted

E_WARNING - Non-fatal errors. Execution is not halted

E_PARSE - Compile-time parse errors.

E_NOTICE - Run-time notices.

E_ALL - All errors and warnings, except level E_STRICT

E_CORE_WARNING - Non-fatal run-time errors. This occurs during PHP's initial start-up.

There are some different locations where we can control errors:

In the php.ini file

In the .htaccess file on your web server

From your own PHP code.

We could do the same from our PHP code during runtime by calling the error_reporting() function as below:

```
error_reporting(E_ALL);
```

To turn on error logging and log the errors to our specific log f ile (instead of the default error log file, which is often the web server error log file)

```
log_errors = on
error_log = "/tmp/error.log"
```

Creating a Custom Error Handler:

We simply create a special function that can be called when an error occurs in PHP. In the example we will send an e-mail with an error message, if a specific error occurs:

```
<?php
//error handler function
function customError($errno, $errstr)
 {
 echo "<b>Error:</b> [$errno] $errstr<br>";
 echo "Error Occured";
 error_log("Error: [$errno] $errstr",1,
 "info@websjyoti.com","From: webmaster@websjyoti.com");
}

//set error handler
set_error_handler("customError",E_USER_WARNING);

//trigger error
$test=2;
if ($test>=1)
 {
  trigger_error("Value must be 1 or below",E_USER_WARNING);
}
?>
```

Explanation:

set_error_handler(): The default error handler for PHP is the built in error handler. We have made the function above the default error handler for the duration of the script. Since we want our custom function to handle all errors,
the set_error_handler() only needed one parameter, a second parameter could be added to specify an error level.

Example
set_error_handler("customError",E_USER_WARNING);
trigger_error() function will raise error when an illegal input occurs.

Using Include() Function:

Suppose that you want to print the same message on every page like your company details and address, or maybe today‘s date. If you are placing the same coding on each page of your site that is not very efficient because-

- You are typing the same information ov er and over again, which is not the good approach.

- In the case of an update or a change, you have to make the change in every single page of your site which is very time consuming process.

A solution to this problem is to use an include() method. You tak e commonly used information and put it in a separate file. For example, if you have a set of variables that need to be use in every page on your site, you could define them once, in a single PHP file. Then, on each of your pages, you can include() it by re ferencing that file.

Difference between Include and require:

The main difference between include and require statements in PHP is that if any error occurred include will raise an error message and then will continue the rest of the code placed in the page where as require will raise a fetal error and terminates execution.

Function require () throws an error (E_COMPILE_ERROR) and stop the script.

Function include () throws a warning (E_WARNING) but the script will continue.

require_once()

function require_once is useful if the result of the code is necessary but subsequent inclusions would throw an error. That goes for scripts with functions for instance. If such a script would be included more than once an error would be thrown since functions cannot be redeclared.

include_once()

Function include_once is useful in the case of including remote files and not wanting them to be included several times due to an HTTP overhead. This is also usefull in payment gateway and process on e-commerce websites.

Examples

```
<html>
<head>
<title>My Web page</title>
</head>
<body>
<div class=”header”>
<?php
```

```
Include_once  “menu.php “;
?>
</div>
<div class=”footer”>
<?php
Include_once  “menu.php “;
?>
</div>
</body>
</html>
```

In this example you will see a warning when you attempt to recall file “menu.php”

PHP Errors - Troubleshooting

Fatal error: Call to undefined function xyz() in localhost/test.php on line 2

Solution: This means that a PHP function that the current page needs cannot be found. check carefully that all of the inclu ded files have been uploaded or replaced and that you have followed the installation instructions.

PHP Warning: include(abc.php): failed to open stream: No such file or directory in localhost/test.php on line 2

Solution: This warning tells you that an include file (included using PHP‘s include syntax) was not found. If you get a warning or error message that a file could not be opened, then check that the

file referred to in the error message is there and that your path to it is correct.

Warning: move_uploaded_file(upload/logo.jpg) [function.move-uploaded-file]: failed to open stream: Permission denied in localhost/upload.php on line 32

Solution: This warning is an example of a permissions error. The script is trying to upload a file to a directory, but the directory doesn‘t have enough permission for the file to be moved to it.

Notes

Working with Images and Other Uploaded Files

Working with uploaded documents is the major part of any server side programing language. Uploading of files such as pdf document, images, video and other docs handled through enctype=”multipart/form-data” attribute in HTML. We need to create a HTML form that allows you to choose the file to be uploaded.

Note: please make sure you have form method type is post and enctype=”mutipart/form-data” to to handle file uploading process successfully.

HTML Code

<form enctype="multipart/form-data" action="process.php" method="POST">

Choose File : <input name="myfile" type="file" />
<input type="submit" value="Upload" />
</form>

Now create a PHP file process.ph p for handling file information Please note you cannot view your uploaded file directly on this page. We will use $_FILES super global to retrieve uploaded file information.

Example: $_FILES['myfile'][‘name’];

$_FILES[][] i s an associate 2 -dimension array and keeps all the information related to uploaded file. The first [] contain name of file field[myfile] and second[] contain predefined attributes[name]

There are some other predefined attributes given below-

$_FILES['file']['tmp_name'] − the uploaded file in the temporary directory on the web server.
$_FILES['file']['name'] − the actual name of the uploaded file.
$_FILES['file']['size'] − the size in bytes of the uploaded file.
$_FILES['file']['type'] − the MIME type of the uploaded file.
$_FILES['file']['error'] − the error code associated with the file.

So let‘s have an example-

```
<?php
// get the uploaded file name
$filename= $_FILES['myfile'][‘name’];
// get the file type
$filetype= $_FILES['myfile'][‘type’];
// get the file size (in bytes)
$filesize= $_FILES['myfile'][‘size’];
echo “<h2>Uploaded file information</h2> “;
echo “File Name : “.$filename;
echo “<br> File Size : “.$filesize;
 echo “<br>File Type : “.$filetype;
?>
```

Once your file is uploaded successfully y ou may add some validations to make it more useful as per your requirement-

```
<?php
// get the uploaded file name
$filename= $_FILES['myfile'][‘name’];
// get the file type
$filetype= $_FILES['myfile'][‘type’];
// get the file size (in bytes)
$filesize= $_FILES['myfile'][‘size’];
// You may put your validations here i.e restrict file size
if($filesize>=1000000)
{
echo “ File is too Large, Maximum file size should be 1 MB”;
exit;
}
// Validation to check if uploaded file is an Image
if($filetype !=”image/jpeg” && $filetype !=”image/png” &&
$filetype !=”image/JPEG”)
{
echo “ Only Images allowed , Kindly upload valid Image.”;
exit;
}
else
```

```
{
// create a folder – upload (in the same location where you have this PHP file) to move uploaded image from your local computer to server
$target = "upload/";
 $target = $target . basename( $_FILES['myfile']['name']) ;
 move_uploaded_file($_FILES['myfile']['tmp_name'], $target);
 echo  " Image Uploaded Successfully";
 }
?>
```

You will be saving your uploaded images to a folder on the web server, which means you need a directory that is writable by the web server. To make the folder writable by the web server, you can use your FTP client for the same.

To write the uploaded image to the target folder, you use the function move_uploaded_file. This PHP function will retrieve the image and move it to the designated location.

Error Messages: PHP returns an error code while uploading a file through HTML Form . We can get the error code during the file upload by PHP. There is some error messages explained below-

UPLOAD_ERR_OK
There is no error, the file uploaded successfully.

UPLOAD_ERR_INI_SIZE

The uploaded file exceeds the upload_max_filesize mentioned in php.ini file.

UPLOAD_ERR_FORM_SIZE

The uploaded file exceeds the MAX_FILE_SIZE directive that was specified in the HTML form.

UPLOAD_ERR_PARTIAL

The uploaded file was only partially uploaded or upload process was interrupted.

UPLOAD_ERR_NO_FILE

No file was uploaded by user. When a user tries to upload a file, but PHP doesn't receive anything.

UPLOAD_ERR_NO_TMP_DIR

Missing a temporary folder

UPLOAD_ERR_CANT_WRITE

Failed to write file to disk

UPLOAD_ERR_EXTENSION

A PHP extension stopped the file uploading process.

Examples

```
// Check if file already exists
if (file_exists($yourfile))
 {
   echo "Sorry, file already exists.";
```

```
 }
// Check if there are any errors in the upload.
if($_FILES['file1']['error'] > 0)
{
   die("An error ocurred ");
}
//Verify uploaded file is an image - Alternate way
if(!getimagesize($_FILES['file1']['tmp_name']))
{
   die("Please ensure you are uploading a valid image.");
}
```

Make Uploaded Images, Your Profile Image

Once you upload an image to the server, you may display it to any section of your webpage. You may do this by firing an Ajax request after you have finished the image upload. Consider the following example-

```
<?php
// put your query along with database connectivity here
$file=mysql_query(“select *  from Users”);

// consider that you have image path in a column called pro_img
```

<img src="<?php echo $file[‘pro_img’] ?>" alt="" height =”300” width=”450” />

This is responsible for showing you the image that has been uploaded. It does this by receiving the name of the file that has recently been uploaded.

PHP File Upload & Security

When you allow users to upload file to the website, you are putting yourself at a security risk. While testing several web applications, we noticed that a good number of well known web applications, do not have secure file upload forms. Some of these vulnerabilities were easily exploited, and we could gain access to the file system of the server hosting these web applications.

Some important factors-

-Check the referrer: Check to make sure that the information being sent to your script is from your website and not an outside sour ce. While this information can be faked, it's still a good idea to check.

-Restrict file types: You can check the mime-type and file extension and only allow certain types to be uploaded.

-Rename files: You can rename the files that are uploaded. In doing so, check for double -barreld extensions like yourfile.php.gif and eliminate extensions you don't allow, or remove the file completely.

-Change permissions: Change the permissions on the upload folder so that files within it are not executable. Your FTP program probably allows you to chmod right from it.

-Block Dangerous Extensions: we encountered file upload forms using a blacklist approach, as a security measure. A list of

dangerous extensions is compiled from the developer, and the access is denied if the extension of the file being uploaded is on the compiled list.

-Client-side validation: Another common type of security used in file upload forms, is client -side validation of files to be uploaded. Typically, such approach is more common in ASP.NET applications, since ASP.NET offers easy to use validation controls

Notes

Managing Sessions & Cookies

Before learning Sessions and Cookies you should know about HTTP and its Stateless process, Stateless means server doesn‘t know about your previous requests as each request is treated new by the browser. So each time when you are browsing your personal account on the website, you have to login again? But we don‘t login again and this is because of Cookies or session.

So, the basic task of both cookies and sessions is to store visitor data so that it can be accessed by every page on a website.

Cookies:

Cookies are small text files that are stored in the visitor's browser for a specific time or for a long -lifespan. Cookies can be edited by the visitor. in short Cookies are client -side files that contain user information.

Sessions:

Sessions are small files that are stored on the website's server. Sessions have a limited lifespan, they expire when the browser is closed or logged out from the session. Sessions cannot be edited by the visitor or user. Session Max life time is 1440 Seconds(24

Minutes) as defined in php.ini file however you may change it accordingly

In short, cookies serve on the visitor's computer and sessions serve on server.

By giving each visitor a cookie with a unique ID, I can use that cookie to recognize each visitor when they return. I can then use sessions to handle the page-to-page data exchange that actually provides each visitor with their customized settings and information, which are provided by each visitor and stored in a database until they are reference by the unique ID stor ed in the cookie.

Cookie Example

PHP allows you to create, retrieve and update cookies. The setcookie() function is used to first create a cookie.

The syntax is: setcookie(name, value, expire, path, domain);

```
<?php
setcookie("username", "Hirdesh Bhardwaj", time()+20);
setcookie("pin", "4412", time()+20);
?>
```

Here we have set a cookie name – username and set value Hirdesh Bhardwaj to it. Cookie username will store this value for next 20 seconds from the current time or the very first when you sent this cookie to browser.

Now, we are free to retrieve the value that is stored in our cookie, using the $_COOKIE super global PHP keyword. It is best to use the isset() function for this, because if a cookie has not been set on the computer that we are trying to retri eve it from, headaches can result.

```
<?php
if (isset($_COOKIE["username"]))
{
echo "Welcome " . $_COOKIE["username"];
}
else
{
  echo "Welcome guest!<br> Cookies Expired.";
}
?>
```

Session Example:

Before using $_SESSION, you have to write session_start(); In that way session will start and you can access $_SESSION variable on that page.

Login System using Session

Let‘s create a program that contain code for the successful login to dashboard by the registered user

Login.html

```
<form method="post" action="login.php">
Email : <input type="text" name="txtEmail" /><br>
Password: <input type="text" name="txtPassword" /><br>
<input type="submit" value="Login" />
</form>
```

Login.php

```
<?php
session_start();
$email=$_REQUEST['txtEmail'];
$password=$_REQUEST['txtPassword'];
/* connecting to the database, you may include a file contain database details */
include 'db.php';
```

```
// Check if username and password are correct
if(mysqli_fetch_array(mysqli_query($con, "select * from student where email ='$email' and password ='$password' ")))
{
// save users email to the session variable.
$_SESSION['email']=$email;
echo "login successfully";
header("Location:welcome.php");
}
else
{
echo "Login failed";
}
?>
```

Welcome.php

```
<?php
session_start();
if(isset($_SESSION[‘email’]))
{
echo $_SESSION['email'];
}
else
```

```
{
echo “This document  is protected”;
}
?>
```

Destroy a Session (Logout)

To remove all global session variables and destroy the session, use session_unset() and session_destroy():

Logout.php

```
<?php
session_start();
session_destroy();
echo “You have logged-out successfully”;
?>
```

Notes:

Sending an Automated email – Mail Function in PHP

Almost in every web application sending automated email by system is the common requirement. So, it‘s very common and important part that must be known by developer.

The simple example, when a visitor to your website fills out a enquiry form.

Mail function returns a Boolean value, return true if email sent successfully, false otherwise

Syntax

mail ($to , $subject , $message, $additional_headers);

$to: This is the first parameter of mail function hold the email address of receiver, the address to which email has to be sent.

$**subject**: Every email should have subject. This parameter cannot contain any newline characters.

$**message**: This is the actual body or content of email to send. Please note that every line should be break by \n and every line should not exceed the limit of 70 characters.

$**addintional_header**: This contain additional header values like From, CC or Bcc etc. This is the optional parameter in mail function.

Example

```
<?php

$to = “info@websjyoti.com”;
```

```
$subject = “Welcome to Webs Jyoti”;

$txt = “Hi, You received a new enquiry from the visitor. ”;

$headers = “From: sales@websjyoti.com” . “\r\n” ;

mail($to,$subject,$txt,$headers);

echo "Mail sent successfully";

?>
```

Example to create enquiry form for the website

Lets create a HTML form first-

```
<form action=”thanks.php” method=”post”>
  Your Name: <input type=”text” name=”name”>
 <br>
 Email : <input type=”email” name=”email”>
 <br>
 Phone: <input type=”text” name=”phone”>
 <br>
 <textarea cols=”40” rows=”7” name=”feedback”></textarea>
 <br>
<input type=”submit” value=”Submit”>
</form>
```

thanks.php

```
<?php
// Collect information
$name=$-REQUEST[‘name’];
$email=$_REQUEST[‘email’];
$phone=$_REQUEST[‘phone’];
$feedback=mysql_real_escape_string($_REQUEST[‘feedback’]);
// Create required parameters for the mail functions
$sendto=”info@websjyoti.com”;
$subject=”New Enquiry on the website”;
$headers= “From: $email” . “\r\n” ;
$message=”----------Enquiry Details----------\n\n Name :”.$name . “\n Email :”.$email. “\n Phone :”.$phone . “\n Feedback “.$feedback ;
```

```
if(mail($sendto,$subject,$message,$headers))
{
echo " Thanks for your feedback. We will contact you soon";
}
else
{
echo " There is some error in sending mail";
}
?>
```

Example to create and send activation code to activate an user profile

Create a code if user is registered successful on the website

```
<?php
$name= $_REQUEST['txtName'];
$email=$_REQUEST['txtEmail'];
$password= $_REQUEST['txtPassword'];
$city= $_REQUEST['txtCity'];
include 'db.php';
// include your mail sending code file (notification.php) here
include_once("notification.php");
```

```
mysql_query("insert into student(name,email,
password,city,`code`)values ('$name', '$email',
'$password','$city','$code')" );

echo “Registration process is done, please check email to verify account to instant activation”;

?>

<script>

alert("You have been register");

window.location='login.html';

</script>

//notification.php

<?php

//$sendto=$email;

$subject="Account Created Successfully- Verify Account";

$code= rand(11111,99999);

$headers ="From:Webs Jyoti info@websjyoti.com \n"; //from address

$headers .= "MIME-Version: 1.0\n";

$headers .= "Content-type: text/html; charset=iso-8859-1 \n";
$mess="Hi, \n\n Thanks for creating account at

<b>Websjyoti.com</b> Your activation code is $code. alternate

you can click here to <a
```

```
href='http://websjyoti.com/temp/verify.php?code=$code'>verify
your account</a>  <br><br> regards, <font color=red'>automail,
webs jyoti</font> ";
$f=mail($email,$subject,$mess,$headers);
if($f == "1")
{
echo "Pls check your mail to activate your account";
}
else
{
echo " pls check mail function";
}
?>
```

//Verify.php

```
<?php

include 'db.php';

$code=$_REQUEST['code'];

mysql_query("update student set status='verified' where
`code`='$code' ");

echo "You have been verified ";

?>
```

Notes:

Object Oriented Programing in PHP

Object-oriented programming (OOP) is an approach, you can broke down your program into several pieces or modules , further can used to creating copies of such modules as per the requirement. It also allows you to group all of the variables and functions of a particular concept into a single class. So Object-oriented programming (OOPs) is more advanced and efficient than the procedural style of programming.

Some advantages of using OOP:

- Reusability- Easy to use and manage code
- Data Hiding – Data Abstraction and Encapsulation
- Modularity
- Resilience to change
- Fast and efficient
- Realistic Modeling

Before we go in details, let‘s define important terms related to Object Oriented Programming.

- **Class** − Class is a collection of member variables and the functions. In other words, a class is a blueprint or a piece of code describing how to manage a task in the way we want

- **Object** −You define a class once and then make many objects that belong to it just like creating multiple houses from a blueprint. Objects can have state behavior and identity. To create an object, we use the ***new*** keyword.

- **Member functions and Variable s** − these are the functions and variables defined inside a class and are used to access object data. Example changing the color or material used, of my house is a method performed on my house object.

- **Inheritance** – Inheritance is defined as creating a new c lass from the existing class. When a class is defined by inheriting existing function of a parent class then it is called inheritance.

- **Polymorphism** – Using this you can create multiple functions with their same name. Or simply saying function name will remain same but it take s different number of arguments and can do different tasks.

- **Overloading** − A type of polymorphism in which some or all of operators have different implementations depending on the types of their arguments. Similarly functions can also be overloaded with different implementation.

- **Data Abstraction** − any representation of data in which the implementation details are hidden (abstracted).

- **Encapsulation** − refers to a concept where we encapsulate all the data and member functions together to form an object.

- **Constructor** – A Constructor is the same name of its class, will automatic invoke once the object of class in created.

- **Destructor** – It refer to a special type of function which will be called automatically whenever an object is deleted or goes out of scope.

Define a PHP *Class and Object*: A class contains set of functions and member va riables (also called properties). A PHP class can hold many functions and variables as follow-

```
class Teacher
{
public $course=”Java”;
function students()
{
//set of statements or lines here
}
};
```

Above class Teacher contains one member variable (Class variable or properties) i.e. *course* and one functions i.e. *students*(). Here *public* is access specified which means variable course can access or used by another class. Please note before declaring a class variable or properties, we should add the visibility ke yword (public / private / protected) to define where the variable is available

Naming a Class: The name you use for a class ("Teacher " in the above example)

- Should not be a PHP reserved word (Keyword).
- Should start with a letter or underscore.
- Can have letters, numbers or underscores after the first letter.

To access member variables or functions of any class you need to create object of that class first. Example below-

```
$ob=new Teacher;
```

Once object of the class is created then you can access member variables or functions something like this-
$ob->students();

echo $ob->course;

Remember to execute or view output from a function you have to call it first. Function will not produce any output until you call it.

What is Constructor? As explained every function should call to get output from the same. But constructor is the function which doesn‘t require any calling as it automatically calls once the object of the class is created. In PHP constructor is defined using __construct() method. Example below-

Example : PHP Constructor

```
class Teacher
{
public $course=”Java”;
function __construct()
{
echo “This function is constructor- default function”;
}
function __destruct()
{
echo “This function is destructor will call after end of program”;
}
};
$ob=new Teacher;
```

In the above program we have created 2 functions __construct() and __destruct() which will automatically called once the object is created. Constructor is used to initialized memory while Destructor is used to release or de -initialized memory of the c lass or usually, it is when the script ends.

Please Note if you have a constructor with arguments, you should send the values for those arguments when you create it.

The $this Keyword

$this is a pseudo -variable which is only available inside methods. $this is used to access properties / member variables of the present class. In other words, we use $this variable to access the current object. Example Below

```
<?php
class Users {
   public $email;
function sendData($x)
{
    $this -> email = $x;
}
};
```

Access specifiers or visibility mode in PHP

There are 3 visibilities or access specifiers available in PHP public, private and protected. These visibilities are used with properties or member variables to enable outside the class. These

specifiers are also p lay important role in validating and restrict information in the program. For example my Credit card and its Pin details are private and I don‘t want to share it with someone else.

- The **Public** properties or methods can be accessed anywhere or outside the class.
- The **Private** properties or methods can only be accessed within the class.
- The **Protected** properties can be accessed by the classes that inherited its parent‘s class.

```
<?php
class myDetails
{
  private pin=”1234”;
  function view_details()
{
    echo $this->pin;      // Inside class this is valid
}
};
 $ob=new myDetails;
 echo $ob->pin;   //Error cannot access outside the class
```

So from the above example, any method inside the class has access to private property (pin). But, you cannot access it from outside the class.

PHP OOPs Inheritance

Inheritance is the process of creating a new class from the existing. However we can also add some additional functionality or resources to the newly created class apart from the existing base class.
It could be useful if you have certain classes represent some common methods or properties. For example The e-Commerce portal have several classes and in each class you require some user information like email and auth key, So instead of creating separate properties for each class you can create a master class and let allow other classes to use these properties using inheritance.

When we say an inherited class is a child class or sub class and from which we inherit is called the parent class or base class.

There are several types of Inheritance in PHP OOPs programing are listed below-

- **Single Inheritance**: When one class is inherited by a single class called Single Inheritance.

- **Multilevel Inheritance:** When a parent class is inherited by a child class then that child class is further inherited by child class.

- **Hierarchical Inheritance**: When a single parent class has more than one child class called hierarchical inheritance.

Please note PHP doesn't support multiple inheritances , However ***Traits*** *are there in PHP to solve the same.*

Example: Single Inheritance in PHP

```
<?php
class A
{
public $var =”Hello”;
function view()
{
echo $this->var;
}
};
class B extends A    // Class B is child and A is parent here
{
function hello()
{
echo $this->var;     // Using properties of parent class A
}
};

$ob=new B;      // Create object of Child class
$ob->hello();
$ob->view();
?>
```

Example: Hierarchical Inheritance in PHP

```
class A
 {
 public $price = 100;
function __construct()    // Parent class constructor
{
echo "This is Parent class";
}
}
```

```
class B extends A   // First child class of A
{
function fn1()
{
$this->price=200;
echo $this->price;
}
}
class C extends A   // Second child class of A
{
function fn2()
{
$this->price=300;
echo $this->price;
}
}
$ob1 = new B;
$ob1->fn1();
echo "<br>";
$ob2 = new C;
$ob2->fn2();
```

Traits in PHP: Since we already know that PHP doesn‘t support multiple inheritances so Traits in PHP can used to declare methods used in multiple classes. By doing so you can reduces code duplication as there's no need to r edefine the same method again and again.

Example: Using Traits in PHP

```
<?php
Trait A
{
public function hello()
{
  echo "Hello This is Trait A";
}
}
Trait B
 {
public function test()
{
  echo "Hello This is Trait B";
}
}
class MyClass
 {
          use A, B;  // Calling Traits A and B in the class
}
$ob = new MyClass();
$ob-> hello();
$ob-> test();  // Calling function test
?>
```

Static Methods and properties in PHP

The static methods are widely used to create Helper and Utility functions (Example autoloading) which gives you more ability to organize and recall your functions. Static functions in PHP can be called directly (Using Resolution Operator ::) without creating the object of the class. The static keyword is used to define static function in PHP.

Example

```
class A
 {
 static function fn()
 {
  echo "This is a Static Function";
}
}
echo A::fn();  // Calling function without creating object
```

Please note if there's no visibility mode or access specifier (Public Private or Protected) is declared then the visibility mode will be ***public*** by default.

Static methods also can accessed from a methods in the same class using the **self** keyword and **::** (Resolution Operator)

Example

```
class A
 {
  public static function test()
 {
  echo "This is static function";
 }
  public function hello()
 {
          self::test();   // Calling a static function using self and ::
}
}
```

Static Properties or Variables: Just like static functions or methods, You can also access static properties or variables in the same way using :: operator. Example

```
<?php
class A
 {
public static $rating = 7;  // Static properties / Member variable
public function __construct()
{
  echo "Rating is" . self::$rating;  // Accessing static variable
}
}
```

```
echo A::$rating;    // Accessing static variable directly
A::$rating = 10;
new A();  // Object to call constructor
```

Getter and Setter in PHP

Getter and Setter methods provide encapsulation of data so that in order to set or get data, it has to pass through a method.

A “getter” allows to you to retrieve or “get” a given property. A “setter” allows you to “set” the value of a given property.

Example

```
<?php

class MyClass
{
  public $greeting = "Good Morning";
  public function setProperty($x)
  {
     $this-> greeting = $x;
  }
  public function getProperty()
  {
```

```
    return $this-> greeting. "<br />";
  }
}
// Creating Class object
$obj = new MyClass;
 // Get the property values
echo $obj->getProperty();
 // Set a new value
$obj->setProperty("Good Afternoon");
// let‘s print the current value
echo $obj->getProperty();
?>
```

Notes

MVC (Model-View-Controller) in PHP:

MVC is an application architecture, or design pattern, whose fundamental principle is based on the idea that the logic of an application should be separated from its presentation. So simply, MVC is a better way of separating the logic of your application from the display. The benefit of using MVC is making your application more maintainable.

The Model:

The model is the most complex of all the parts of the system and will contain all the logic which is specific to the application. It is the part of the application which takes data (from any source) and processes it. So simply, the model is responsible to manage the data; it stores and retrieves entities used by an application, usually from a database, and contains the logic implemented by the application.

The View

The view contains all the display logic. In PHP it will be the part of the application which generates the HTML. It has direct access to the Model and can query the model to get its data. So simply, the view is responsible to display the data provided by the model in a specific format. It has a similar usage with the template modules present in some popular web applications, like WordP ress, Joomla etc.

The Controller

The controller takes user input and upda tes the model where required. Please note that the controller is not a mediator or gateway between the view and the model. The view gets its own data from its model. The controller accesses the model but does not contain any display logic itself. All the c ontroller does is respond to user input.

In other words, "The controller handles the model and view layers to work together. The controller receives a request from the client, invokes the model to perform the requested operations and sends the data to the View. The view formats the data to be presented to the user, in a web application as an html output."

MVC Program flow:

-The model, view and controller are initialized

-The view is displayed to the user, reading data from the model

-The user interacts with the view (e.g. presses a button) which calls a specified controller action

-The controller updates the model in some way

-The view is refreshed (retrieving the updated data from the model)

A Simple MVC Program:

```
<?php
class Model
 {
    public $text;
    public function __construct()
 {
     $this->text = 'Hello.. Lets learn MVC pattern in PHP';
    }
}
class View
{
    private $model;
   public function __construct(Model $model)
 {
        $this->model = $model;
   }
public function output()
 {
     return '<h1>' . $this->model->text .'</h1>';
```

```
  }
}
class Controller
{
 private $model;
 public function __construct(Model $model)
   {
     $this->model = $model;
    }
}
$model = new Model();
$controller = new Controller($model);
$view = new View($model);
echo $view->output();
?>
```

Notes

Mysql/Maraidb – Advanced Learning

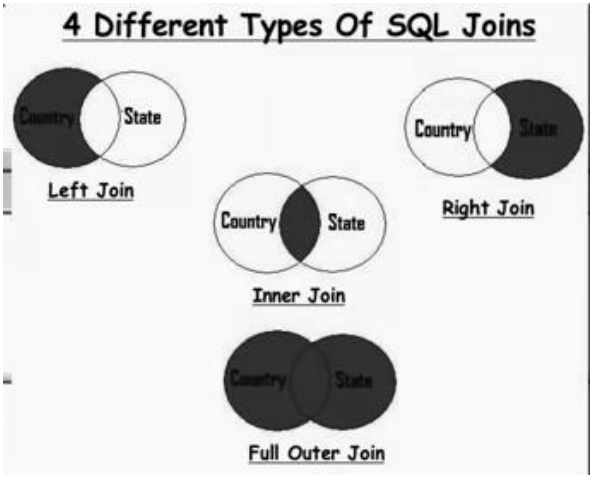
Joins: MYSQL JOINS are used to retrieve data from multiple tables. In most real world MySQL usage, you will often need to get data from multiple tables in a single query. So to access data from different tables, that may relate to each other, we use the JOIN clause in MySQL

There are three types of MySQL joins:

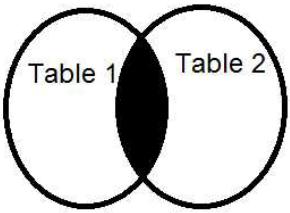
INNER JOIN (Simple join)

LEFT OUTER JOIN (LEFT JOIN)

RIGHT OUTER JOIN (RIGHT JOIN)



INNER JOIN OR Simple join: The MYSQL INNER JOIN is used to return all rows from the tables where the join condition is satisfied. It is the default join also. So in simple words it can be used to retrieve common records / rows from the multiple tables.



Consider two tables " users" and " rec", having the following data.

Table - Users

+ Options

	uid	email	city	name
Edit Copy Delete	1	h@gmail.com	gurgaon	hirdesh
Edit Copy Delete	2	suzeet@gmail.com	delhi	sujeet
Edit Copy Delete	3	pooja@gmail.com	noida	pooja
Edit Copy Delete	4	sw@gmail.com	gurgaon	swati

Table - rec

+ Options

	uid	month	days
Edit Copy Delete	1	Jan	23
Edit Copy Delete	2	jan	21
Edit Copy Delete	2	march	30
Edit Copy Delete	3	feb	25
Edit Copy Delete	4	jul	21
Edit Copy Delete	3	mar	12
Edit Copy Delete	4	dec	22
Edit Copy Delete	4	jul	13

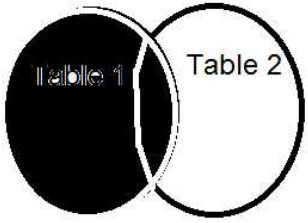
This MySQL INNER JOIN example would return all rows from the users and rec tables where there is a matching uid value in both the tables.

Query:

Select * from users inner join rec on users.uid=rec.uid;

LEFT OUTER JOIN:

MYSQL LEFT OUTER JOIN returns all rows from the LEFT - hand table specified in the ON condition and only those rows from the other table where the joined fields are equal.

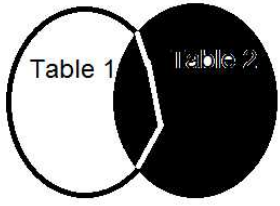


This would return all rows from the users table and only those rows from the rec table where the joined fields are equal.

Query:
Select * from users left join rec on users.uid=rec.uid;

RIGHT OUTER JOIN

MYSQL RIGHT OUTER JOIN returns all rows from the RIGHT - hand table specified in the ON condition and only thos e rows from the other table where the joined fields are equal.



Query:

Select * from users right join rec on users.uid=rec.uid;

This would return all rows from the rec table and only those rows from the users table where the joined fields are equal.

Assignments for the practice

1. Write a SQL statement to prepare a list with Customer name, City,cities,PLAN TYPE, Payment mode and their Income Sources for the customers who are present in both tables.

2. Write a SQL statement to prepare a list with Custo mer name, City,cities,PLAN TYPE, Payment mode and their Income Sources for the customers who are present in both tables and their policy status is active.

3. Write a SQL statement to make a list with customer name, Deposit amount and their cities for those orders which deposit amount between 7000 and 10000

4. List of customer who are not there in secondary table but not in the primary table (hint: using right join on Email address)

MYSQL Indexes : A MYSQL index is a data structure that improves the speed of operations in a table. Basically, Indexes are used to sort data to make it easier for searching operation.

It works in the same way as the indexes you see in this book you are reading. So the SQL Query without scanning each row or entire table it quickly jumps to an index. You can create index on the column(s), you use frequently for searching records using where clause. A table or views can have more than one index built from it.

In order to create an index, below is the syntax-

CREATE INDEX index_name ON table_name;

Example:

Create index emp_salary on employees (Salary ASC);

Above query will order salary column values in Ascending order and now if you will use where query on Salary column it will execute faster as you have already shorted column through the index key. Example

*select * from employees where Salary>10000 and Salary <20000;*

Above index is non NON CLUSTERED type, which is by default in SQL. it only short records in a heap for fast searching purpose

and not in the actual table, so the co lumn values will remain in the same place even after applying non clustered index.

You can have multiple non clustered indexes in the same SQL table.

Create Nonclustered index emp_sa lary on employees (Salary ASC);

Query To view index in a table

show index from employee;

Query To drop index command-

drop index employees.emp_salary;

Please note indexes (Type Clustered Index) are automatically created when you add primary or unique key to a column, as Primary Key itself is a Clustered index. So the process of automatically index creation is called implicit indexes

Clustered Index: When you are adding up a primary key to a column that means you have already added three constraints i.e P**rimary key**, **Not null** and **Clustered index**. So a table can have only clustered index.(however that one clustered can refer to multiple column called composite index)

You can also create clustered index with primary or without primary key (Also called custom indexing) as below-

create clustered index ids on employees (id desc);

create a composite clustered on the multiple tables

create clustered index myIndex on employees (id desc,gender ASC);

Non-clustered vs clustered Index: Clustered index helps in arranging the rows physically in the memory . So the changes after index can be seen in the table and not in the heap. On the other hand, Non clustered index will not arrange rows physical in the memory in sorted order.

Please Note The MySQL InnoDB engine has mandatory clustered indexes. That means there is always a clustered index, often using the primary key. If there is no suitable unique key available, MySQL will use a generated row ID for that purpose. The MyISAM storage engine doesn't support clustered indexes and uses heap-tables all the time.

Only one type of index (Non-clustered) can be used in the MYSQL Engine.

Notes

Mysql – Stored Procedure:

Typically stored procedures help increase the performance of the applications. Once created, stored procedures are compiled and stored in the database. However MySQL implements the stored procedures slightly different. MySQL stored procedures are compiled on demand. After compiling a stored procedure, MySQL puts it to a cache.

Stored procedure helps reduce the traffic between application and database server because instead of sending multip le lengthy SQL statements, the application has to send only name and parameters of the stored procedure.

Stored procedures are secure. Database administrator can grant appropriate permissions to applications that access stored procedures in the database wi thout giving any permission on the underlying database tables.

Note: do not create so many stored procedures as they took lot of memory.

Example

```
DELIMITER //

CREATE PROCEDURE myproc()

BEGIN

SELECT *  FROM users where city='gurgaon';

END //
```

```
DELIMITER ;

//To Execute or Call procedure myproc

call myproc();
```

Explanation:

-We use the delimiter to indicate the start or end of the stored procedure. The last command (DELIMITER;) changes the delimiter back to the semicolon (;).

-We use the CREATE PROCEDURE state ment to create a new stored procedure. We specify the name of stored procedure after the CREATE PROCEDURE statement. We put the parentheses after the name of the stored procedure.

-The section between BEGIN and END is called the body of the stored procedur e. You put the declarative SQL statements in the body to handle business logic.

Store procedure with parameters

```
DELIMITER //

CREATE PROCEDURE proc(IN city1 VARCHAR(50))

 BEGIN

 SELECT *  FROM users WHERE city = city1;

 END //

DELIMITER ;
```

To Execute Procedure proc

call proc ('delhi');

MySQL Loop in Stored Procedures

MySQL provides loop statements that allow you to execute a block of SQL code repeatedly based on a condition. There are three loop statements in MySQL: WHILE, REPEAT and LOOP.

While Loop Example

```
DELIMITER $$

CREATE PROCEDURE myloop()

 BEGIN

 DECLARE x  INT;

 DECLARE str  VARCHAR(255);

 SET x = 1;

 SET str =  '';

 WHILE x  <= 10 DO

 SET  str = CONCAT(str,x,',');

 SET  x = x + 1;

 END WHILE;

  SELECT str;

 END$$
```

DELIMITER ;

First, we build str string repeatedly until the value of the x variable is greater than 10. Then, we display the final string using the SELECT statement.

So Let‘s Call to myloop procedure-

CALL myloop();

Output: 1 2 3 4 5 6 7 8 9 10

MYSQL – Triggers

A SQL trigger is a s et of SQL statements stored in the database. it is fired whenever an event associated with a table occurs e.g., insert, update or delete.

The main difference between a trigger and a stored procedure is that a trigger is called automatically when a data modification event is made against a table whereas a stored procedure must be called explicitly.

It is important to understand SQL trigger‘s advantages and disadvantages so that you can use it appropriately.

Advantages of using SQL triggers

SQL triggers prov ide an alternative way to check the integrity of data.

SQL triggers provide an alternative way to run scheduled tasks.

SQL triggers are very useful to audit the changes of data in tables.

Disadvantages of using SQL triggers

SQL triggers only can provide an extended validation and they cannot replace all the validations.

SQL triggers are invoked and executed invisible from the client applications; therefore, it is difficult to figure out what happen in the background.

Two types of trigger

1. Before Trigger - Before Trigger fires before the execution of INSERT, DELETE and UPDATE commands.

2. After Trigger - After trigger fires after the execution of INSERT, DELETE and UPDATE commands.

We are going to create a table called customer having two columns to prepare a log whenever a new record is added or deleted in the same table

- table logfile will insert the current time whenever new value add/delete

--DELIMITER $$ for avoid ; on every end of line all statements treats as group and continue line until DELIMITER $$ closed

```
DELIMITER $$

CREATE TRIGGER trigger1 BEFORE INSERT ON users

FOR EACH ROW BEGIN

insert into logfile(lastupdate) values(now());

END$$

DELIMITER ;
```

DROP A TRIGGER

DROP TRIGGER [IF EXISTS] [schema_name.]trigger_name

This statement drops a trigger. The schema (database) name is optional. If the schema is omitted, the trigger is dropped from the default schema. DROP TRIGGER was added in MySQL 5.0.2. Its use requires the TRIGGER privilege for the table associated with the trigger.

Notes

MySQL Transactions:

A transaction is a sequential group of database operations or queries, which is performed as if it were one single work unit. In other words, a transaction will never be complete unless each individual operation within the group is successful. If any operation within the transaction fails, the entire transaction will fail.

So in short you will club or combine multiple SQL queries into a group and you will execute all of them together as the part of a transaction.

Why transaction is required in database:

Consider a n example of ATM Machine. We all use to withdraw and transfer money by using ATM. If you break withdrawal operation into individual steps you will find:

1) Verify account details.
2) Accept withdrawal request
3) Check balance
4) Update balance
4) Dispense money

Suppose your account balance is 1000$ and you make a withdrawal request of 900$. At fourth step, your balance is updated to 900$ and ATM machine stops working due to power outage

Once power comes back and you again tried to withdraw money you surprised by seeing your balance just 100$ instead of 1000$. This is not acceptable by any person in the world :) so we need a transaction to perform such task. If SQL statements would have been executed inside a transaction in database balance would be either 100$ until money has been dispensed or 1000$ if money has not been dispensed.

In the context of transaction processing, the acronym ACID refers to the four key properties of a transaction: atomicity, consistency, isolation, and durability.

Atomicity
A transaction should be completed in its entirety or not at all. If a transaction aborts in the middle, all operations up to that point should be undone completely.

Consistency
A transaction should transform a database from one consistent state to another consistent state.

Isolation
Each transaction should occur independently of other transactions occurring at the same time.

Durability
Committed transactions should be fully recoverable in all but the most extreme circumstances. Write-ahead logs provide absolute data durability until data is eventually written into permanent data and index files.

Example: To implement the above ATM machine example for transfer 900$ from Account No. 9001 to Acc. No. 9002

```
start transaction
select balance from Account where Account_Number='9001';
select balance from Account where Account_Number='9002';
update Account set balance=balance-900  here
Account_Number='9001' ;
update Account set balance=balance+900  here
Account_Number='9002' ;
commit;    // if all sql queries succeed
rollback;  // if any of Sql queries failed or error
```

Another Example:

```
<?php
// required php 5+ version as its latest concept of PHP
// switch db to innoDB while creating table as this support acid
$mysqli = new mysqli("localhost", "my_user", "", "hirdesh");

/* check connection */
if (mysqli_connect_errno())
 {
   printf("Connect failed");
   exit();
}

/* set autocommit to off */
$mysqli->autocommit(FALSE);
```

```
/* Insert some values */
$mysqli->query("INSERT INTO users VALUES ('hirdesh',
'gurgaon', 'M', 11.2)");
$mysqli->query("INSERT INTO users VALUES ('ashok',
'gurgaon', 'M', 5.2)");

/* commit transaction */
if (!$mysqli->commit())
 {
   print("Transaction commit failed\n");
   exit();
}
/* drop table */
$mysqli->query("DROP TABLE users");

/* close connection */
$mysqli->close();
?>
```

Example 2:

```
<?php
// another example by steps
try {
   // First of all, let's begin a transaction
   $db->beginTransaction();
 // A set of queries; if one fails, an exception should be thrown
```

```
    $db->query('first query');
    $db->query('second query');
    $db->query('third query');

    // If we arrive here, it means that no exception was thrown
    // i.e. no query has failed, and we can commit the transaction
    $db->commit();
    }
 catch (Exception $e)
 {
    // An exception has been thrown
    // We must rollback the transaction
    $db->rollback();
}
?>
```

Notes

Ajax (Asynchronous Java script And XML)

Introduction

In Jesse Garrett‘s original article that coined the term, it was AJAX. The “X” in AJAX really stands forXMLHttpRequest though, and not XML. Jesse later conceded that Ajax should be a word and not an acronym and updated his article to reflect his change in heart. So “Ajax” is the correct casing. As its name implies, Ajax relies primarily on two technologies to work: JavaScript and the XMLHttpRequest. Standardization of the browser DOM (Document Object Model) and DHTML also play an important part in Ajax‘s success, but for the purposes of our discussion we won't examine these technologies in depth.

How Ajax Works

At the heart of Ajax is the ability to communicate with a Web server asynchronously without taking away the user‘s ability to interact with the page. The XMLHttpRequest is what makes this possible. Ajax makes it possible to update a page without a refresh. By Ajax, we can refresh a particular DOM object without refreshing the full page. Let's see now what actually happens when a user submits a request:

1. Web browser requests for the content of just the part of the page that it needs.
2. Web server analyzes the received request and builds up an XML message which is then sent back to the Web browser.

3. After the Web browser receives the XML message, it parses the message in order to update the content of that part of the page.

The XMLHttpRequest object is part of a technology called Ajax (Asynchronous JavaScript and XML). Using Ajax, data could then be passed between the browser and the server, using the XMLHttpRequest API, without having to reload the web page.

The XMLHttpRequest Object: In order to make a request to the server through Ajax, an object must be created that can be used for different forms of functionality. Microsoft Internet Explorer creates the object as an ActiveX control, whereas browsers such as Firefox and Safari use a basic JavaScript object.

Popular XMLHttpRequest Methods:

abort(): The abort method is used to stops the request in its tracks. If you only want a request to fire for a certain length of time, you can call the abort method to stop the request prematurely.

getAllResponseHeaders() : You can use this method to obtain the full information on all HTTP headers that are being passed. Examples - Server name, Date, Connection & Content Type.

open ("method","URL","async","username","pswd"):This is the method you use to open a connection to a particular file on the

server. It is where you pass in the method to open a file (GET or POST), as well as define how the file is to be opened.

setRequestHeader("label","value") : You can give a header a label of sorts by passing in a string representing both the label and the value of said label. this method may only be invoked after the open() method has been used, and must be used before the send function is called.

send(" "): This is the method that actually sends the request t o the server. If the request was sent asynchronously, the response will come back immediately; if not, it will come back after the response is received.

XMLHttpRequest Properties: Some major XMLHttpRequest() properties are as follow-

onreadystatechange: The onreadystatechange property is an event handler that allows you to trigger certain blocks of code, or functions, when the state is changed.

readyState: The readyState is useful property for exception handling, and can be important when deciding when to perform certain actions. For example, you could have a set of code execute when readyState is loading, or stop executing when readyState is complete.

responseText: The responseText property will be returned once a request has gone through. If you are rais ing a request to a script of some sort, the output of the script will be returned through this property.

Once a request has been received, the server then decides what response to return. There are many different response codes as follow.

200 - OK

401 Unauthorized

403 Forbidden

404 Not Found

500 Internal Server Error

Sending a Request to the Server:

In order to make a request to the server you need to decide what page (or script) you want to connect to, and then what area to load the page or script into.

Let‘s have a basic example-

<script>

function sendRequest(serverPage, objID)

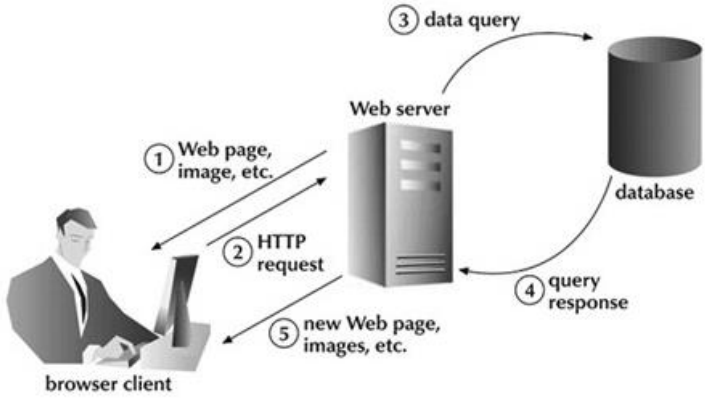
{

```
var obj = document.getElementById(objID);

xmlhttp.open("GET", serverPage);

xmlhttp.onreadystatechange = function()

{

if (xmlhttp.readyState == 4 && xmlhttp.status == 200)

{

obj.innerHTML = xmlhttp.responseText;

}

}

xmlhttp.send(null);

}

</script>
```

Explanation: The code here is taking in the HTML element ID and server page. It then attempts to open a connection to the server page using the open() method of the XMLHttpReques t object. If the readyState property returns a 4(complete) code and the status property returns a 200 (OK) code, then you can load the response from the requested page (or script) into the innerHTML element of the passed-in object after you send the request.

Steps of AJAX Operation

- A client event occurs.
- An XMLHttpRequest object is created.
- The XMLHttpRequest object is configured.
- The XMLHttpRequest object makes an asynchronous request to the Webserver.
- The Webserver returns the result containing XML document.
- The XMLHttpRequest object calls the callback() function and processes the result.
- The HTML DOM is updated.



Sample Code

<script>

function showme(str)

{

```
// to checking if you don‘t enter anything...

if(str.length==0)

{

  document.getElementById("result").innerHTML='pls type
something‘;

  return;

 }

// make xmlhttp object for request

if (window.XMLHttpRequest)

  {

  // code for IE7+, Firefox, Chrome, Opera, Safari

  xmlhttp=new XMLHttpRequest();

  }

else

  {// code for IE6, IE5

  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

  }

xmlhttp.onreadystatechange=function()

  {


document.getElementById("result").innerHTML=xmlhttp.respons
eText;

  }
```

```
  // send your request to getlist.java page via GET method with
argument city

 xmlhttp.open("GET","getlist.php?city="+str,true);

xmlhttp.send();

}

</script>

<form>

<input type="text" onkeyup="showme(this.value)" name="city" />

</form>
```

here you will get your result in result id by getelementbyid method..

```
<span id="result"></span>
```

Getlist.php

```
<?php

// Receive request on php page

$city=$_REQUEST[‘city’];

echo "City is “.$city;

// You may collect data from database on the base of requested city

?>
```

Notes

Hosting Your Website

Many people like to run their own servers out of their homes or offices, and that is a fe asible solution for hosting, if you have the time. But hosting your own Web site can lead to more problems than it‘s worth.

You need to think about backup power, keeping the security holes patched regular maintenance, regular upgrades, and many other issues. And keep in mind that not only do you need to have a Web server running; you need to have something to take care of your domain name

Servers (DNS Servers)

A server is a just like your computer that stores files and serves when requested.

Server stays up and running 24 hours a day. It‘s their job to make sure your site is secure and always available for viewing. For example when you write a blog or any post, you save the file to your web host's server and now it can be accessed by anyone visiting your blog.

Domain name:

The domain name is the most important part of a web address. A domain is the identity of the company on the internet. Each domain has their extension followed by. Operator For example www.irctc.gov.in is the .gov.in domain means it is a government website. Some popular extensions are .com / .co. / .in / .net /.me

Web Hosting

A web host (web hosting company) helps put you website online and accessible to the public. Here's what a web host allows you to do:

- Gives you space to save your files, images and other media

- Allows people to access your stored files/images/media when requested.

When making your hosting choice. Here are a few criteria to look at when outsourcing hosting:

- Supported languages: PHP, JAVA, CGI, and so on.

- Supported databases: MySQL, Postgresql, MSSQL, and so on.

- Server control: Super User Access

- Server access: Such as FTP, telnet, SSH, and so on..

- Administration GUIs: E-mail, database, user setup, and so on.

- Bandwidth usage: Web site, e -mail, streaming m edia, database connections, and so on.

Types Of Web Hosting: Almost all hosting companies offer these three popular hosting options for you-

-Shared hosting.

-Reseller hosting.

-VPS and Dedicated hosting.

Shared Hosting: Under this plan you will be able to h ost many websites some Shared Hosting plans allow you to host unlimited websites also. In the shared hosting plan You get only one CPanel (Control panel) to manage all the websites.

Reseller Hosting: Reseller hosting allows you to resell your hosting to ot her customers. But you can also use it to host your own websites. In Reseller Hosting account you can create a separate account for each website. So When you buy a Reseller hosting package, you get a Cpanel (Control panel) and WHM (Web Hosting Manager) for creating managing different website accounts.

VPS and Dedicated hosting: PS hosting are better suited for very high traffic sites and sites that require a lot of functionality.Typically, a web host will not allow you to consume so much resources if you ar e on a shared or reseller hosting plan. They would disable your site and ask you to move to a VPS or dedicated plan.

Storage Space and Bandwidth: As mentioned above, a web hosting company gives you space to store your files bandwidth to make your files accessible to public. So basically, Storage Space is the space to store your blog posts, images, media files (videos, podcasts etc.), WordPress installation, themes, plugins etc. are all stored in the server.

Bandwidth: Each time a visitor accesses your websi te, he/she uses server resources. This is because; the server needs to send all the files/images/media to his computer.

So the resources used to serve files this way are calculated as bandwidth.

The more visitors you have, the more server resources will be required and hence you will need more bandwidth.

Bandwidth is measured in Bytes and the values will generally be given in GB (Gigabytes) or TB (Terrabyes).

Supported Languages

First of all, we will talk about the supported languages. Obviously, because you bought this book, we're assuming you are looking into using PHP, but there are other languages you may need to use. There may be a time when another language, such as JavaScript, is better suited for a job than PHP. For example,

if you want live stream ing audio or video on your site, you are likely to use a Java applet of some sort because PHP needs a request from the server to process information. PHP can do it with different calls to the server, but because it is more of a client request from the browser, a client-side language works much better for this task.

There may also be times when you need to use another programming language to accomplish something a client already has set up at a different host or server. If so, it is nice to at least have the option of using, say, a Perl script, rather than spending the time and money to redevelop the application in PHP.

Supported Databases

Again, because this book is geared toward MySQL, you will probably need a host that supports MySQL. However, there are many other databases you can use with PHP.

Here are just some of the databases that PHP can work with:

- MySQL

- PostgreSQL

- MS SQL Server

- MS Access

Depending on your situation, you may want to choose a host that has more than one of these databases set up by default. Some larger companies, for example, are using MSSQL as their database, usually because they are using ASP (Active Server Pages from Microsoft) for their programming. Should you need to convert any site to PHP, you will be glad to know that PHP can connect and work nicely with MSSQL as well. Also, keep in mind that you don‘t have to continue using the other databases; you can always port the data over to MySQL using PHP to ease the troubles of manual conversion.

A MySQL Database is required i f you plan to run a site using a CMS like WordPress or Opencart.

A CMS like WordPress uses the MYSQL database to store and retrieve data to dynamically generate webpages.

A database is not required if you plan to run a static website without using any CMS or PHP language.

Server Control and Access

Many hosts out there won‘t give a Web developer full access and/or control over their hosted domain.

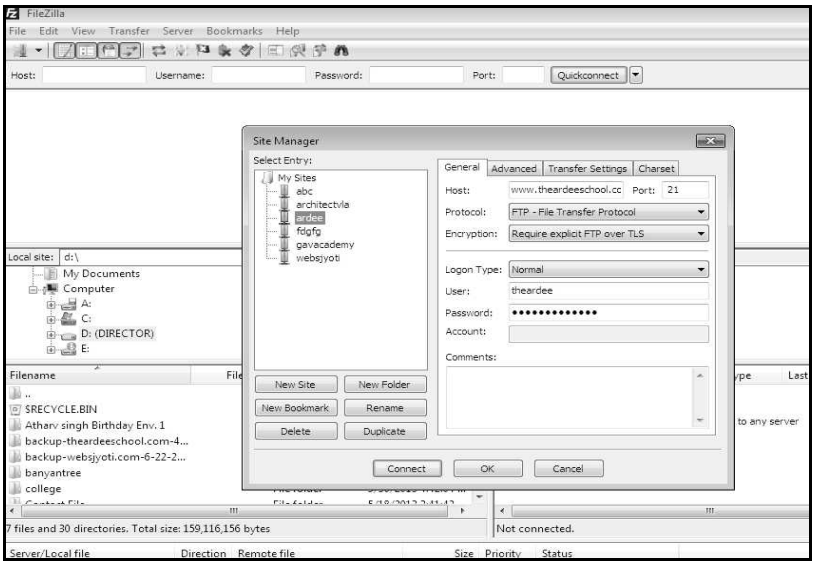
We tend to shy away from those hosts because you are more likely to run into problems with them when you want to do some custom configuration to the server.

Look into the type of access your host provides. Obviously, your host will give you FTP access so you can upload your files to the Web server. Some hosts, however, will give you FTP access but nothing else.

The problem is that you are likely to run into a situation in which you want to configure your server. For

This you will need either telnet or SSH access to use the command line.

In fact, the ability to configure is often necessary when performing tasks that usually aren‘t allowed by hosts by default. For example, consider htaccess. With htaccess, you can deny and allow access to certain files and directories based on the users you allow in the htpasswd file



Administration GUIs

Certain hosts offer Administr ation GUIs (Graphical User Interfaces) or User Control Panels as a feature of their packages. A lot of people don‘t really care for GUIs, but when you don‘t have a choice, either because you don‘t have sufficient access to the server or you don‘t fully und erstand how to get things done through telnet, a point-and-click solution is a wonderful tool.

The interface can be as simple as one that allows you to view information about the server, or it can be as complex as one that allows you to install application s and programming languages with the click of a button. Also, keep in mind that if you have a client that wants to be able to administer its e -mail users, many of the control panels have utilities that allow the client to do so themselves. With such a feat ure, rather than having to call you or

the hosting company to set up an e -mail account, the client can simply log on to the control panel and set up and delete users as the need arises.

Notes

Notes

PHP Pagination

When we display large amount of c ontent on the same page then we start thinking about adding pagination because loading all data on the same page is not the right choice in terms of optimization. So we split all our content to several pages. And in this case – each page contain some part of our information.
And we can just use user -side as well as client side pagination using javascript and PHP to manage with our pages.

It‘s always possible that your SQL SELECT statement query may result into thousand of records. But it is not good idea to display all the results on one page. So we can divide this result into many pages as per requirement.

Paging means showing your query result in multiple pages instead of just put them all in one page.

A simple example to fetch records using LIMIT to ge nerate paging

```
<?php

include 'dbconnectivity.php';

// to check if page url already contain page number like
test.php?pageNum=2

if (isset($_GET['pageNum']))

  {

  $pageNum = $_GET['pageNum'];

} else
```

```
{

// if no pageNum variable exist in url then set pageNum variable to
page 1

  $pageNum = 1;

}

// place your query here

$query=mysql_query("select * from imageslave");

$resultRows = mysql_num_rows($query);

// Set number of Records per page

$rowsPerPage = 4;

/* Calculate how many pages to be generate example if you have
20 record and you want to display 4 pages each page then last page
will be 5.  ( 20/4 =5) */

$lastPage = ceil($resultRows/$rowsPerPage);

// set page position to beginning or at the end of the page

if ($pageNum > $lastPage)

 {

  $pageNum = $lastPage;

   }

    else if ($pageNum < 1)

{
```

```
$pageNum = 1;
}
while($row =mysql_fetch_assoc($query))
{

// display data fields here
echo $row['fname'];
echo " " .$row['lname']."<br>";
}
echo "<br><br>";
// Display links of Next / Prev Page
if ($pageNum > 1)
 {
   $prevPage = $pageNum-1;
   echo " <a
href='{$_SERVER['PHP_SELF']}?pageNum=1'>FIRST</a> ";
   echo " <a
href='{$_SERVER['PHP_SELF']}?pageNum=$prevPage'>PREVI
OUS</a> ";
}
echo "Page $pageNum of $lastPage ";
if ($pageNum < $lastPage)
```

```
{

  $nextPage = $pageNum+1;

  echo " <a
href='{$_SERVER['PHP_SELF']}?pageNum=$nextPage'>NEXT<
/a> ";

  echo " <a
href='{$_SERVER['PHP_SELF']}?pageNum=$lastPage'>LAST</
a> ";

}

?>
```

Notes

Bootstrap – Framework

Bootstrap is a free and open -source CSS framework dir ected at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. *(Source: Wikipedia)*

Originally named Twitter Blueprint, was developed by **Mark Otto and Jacob Thornton** at **Twitter** as a framework to encourage consistency across internal tools.

Installation Guide : You can visit getbootstrap.com to download the latest version of bootstrap. After downloading the setup you need to unzip or extract it. You will get three directories as below

CSS – Contain required css files example – bootstrap.min.css, bootstrap.theme.css etc

Fonts - Contain tft and rtf files for additional fonts

JS - Contain required JS Files – example – boostrap.min.js, npm.js etc.

Sample Code: Installing Bootstrap

```
<!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 101 Template</title>

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

</head>

<body>

// Your HTML code goes

</body>

</html>
```

Some Major Classes available in Bootstrap 3

.container

- sets fixed width to an element (which cha nges depending on a screen size to other fixed values, so it's still responsive) on all screen sizes except xs - on xs, the width is calculated automatically (this behaviour can be changed)
- a fixed width responsive website should be wrapped in .container (or just the content without menu, whatever - it's up to you)
- .container can't be nested!

.container-fluid

- sets 100% width, margin-left and margin-right: auto, padding-left and padding-right: 15px
- a full screen website should be wrapped in .container - fluid
- .container-fluid can't be nested!

.row

- creates horizontal groups of columns (which usually have width classes, see below)
- must be within .container to ensure correct padding and alignment
- only columns (block elements with width classes) may be immediate children of rows
- sets margin-left and margin-right: -15px

.pull-right
.pull-left

- sets float to either left or right

.center-block

- sets margin-left and margin-right to auto

Bootstrap Grid Classes

- can be nested, the nested columns must be wrapped in .row as well
- all columns have float: left and position: relative styles

.col-xs-1 .col-xs-2 .col-xs-3.col-xs-4.col-xs-5.col-xs-6.col-xs-7.col-xs-8.col-xs-9.col-xs-10.col-xs-11 .col-xs-12

- sets width of a column (specified in %) on extra small screens (<768pxt)

.col-sm-1.col-sm-2.col-sm-3.col-sm-4.col-sm-5.col-sm-6.col-sm-7.col-m-8.col-sm-9.col-sm-10.col-sm-11.col-sm-12

- sets width of a column (specified in %) on small screens (≥768px)

.col-md-1.col-md-2.col-md-3.col-md-4.col-md-5.col-md-6.col-md-7.col-d-8.col-md-9.col-md-10.col-md-11.col-md-12

- sets width of a column (specified in %) on medium size screens (≥992px)

.col-lg-1.col-lg-2.col-lg-3.col-lg-4.col-lg-5.col-lg-6.col-lg-7.col-lg-8.col-g-9.col-lg-10.col-lg-11.col-lg-12

- sets width of a column (specified in %) on large screens (≥1200px)
- sets offset (margin-left) by specified number of columns (eg. set .col -sm-offset-3 class to a column to move it by three columns on small screens)

.col-xs-pull-* (.col-xs-pull-1, .col-xs-pull-2 ... up to -12)
.col-sm-pull-* (.col-sm-pull-1, .col-sm-pull-2 ... up to -12)
.col-md-pull-* (.col-md-pull-1, .col-md-pull-2 ... up to -12)
.col-lg-pull-* (.col-lg-pull-1, .col-lg-pull-2 ... up to -12)

- sets left position in % as it is with columns (e.g. left: 75% with -pull-9)

Bootstrap Visibility Classes

- you can apply all as many of these classes on a single element as you wish and this mixing gives you even more possibilities for your layout

.clearfix

- obvious, clear: both
- often combined with classes below

.visible-xs-block and .visible-xs
.visible-sm-block and .visible-sm
.visible-md-block and .visible-md
.visible-lg-block and .visible-lg

- displays element (sets display: block property) only on screens with specified size and hides it (sets display: none) on screens of other sizes automatically
- often combined with .clearfix so that clearfix is effective only on the selected screen size

.hidden-print

- hides element (sets display: none) in print (pre)view
-

Bootstrap Typography Classes

- default font-size of <body> a <p> elements is 14px, line-height 1.428; <p> also has margin -bottom of half of its line-height (20px with 14px font), therefore 10px by default
- all elements like b, big, code, em, i, mark , small, strong, sub, sup, ins, del, s, u, abbr, address and blockquote (and small and cite inside) are properly styled
- vast majority of these classes are most often set to either paragraph, strong or em elements

.page-header

- class for a wrapping div of a heading
- adds some margin, padding and border-bottom

.lead

- for highlighting an element

- sets element's font-size to 21px, font-weight: 300, line-height: 1.4 and margin-bottom: 20px

.text-left .text-center .text-right .text-justify

- aligns texts, usually paragraphs, with text-align CSS property

.text-nowrap

- keeps the whole text with this class on a single line, no wrapping occurs because of white-space: nowrap CSS property of this class

.text-lowercase .text-uppercase .text-capitalize

- sets text-transform CSS property to make all characters inside the element with this class lowercase, uppercase or capitalized

.text-muted

- makes text grey

.text-primary

- makes text blue

.text-success

- makes text green

.text-info

- makes text dark blue

.text-warning

- makes text dark yellow

.text-danger

- makes text red

Bootstrap Lists' Classes

.list-unstyled

- removes list-style and margin-left on list items
- doesn't apply for nested lists (if needed, the nested list has to have .list-unstyled class as well)

.list-inline

- places list items on a single line (with display: inline - block and padding)

.dl-horizontal

- class for a definition list (<dl>)
- sets nice styles for <dt> and <dd> within the definition list (text align, font-weight)

.list-group

- class for the parent elem ent (usually <ul>, but can be as well) of list items
- list items have to have .list-group-item class

.list-group-item

- class for .list-group children elements
- for a nicer look of the list items
- usually on <li>, but can be <a> as well

.list-group-item-heading

- for a heading within .list -group-item, often <h4> or similar low-level heading

.list-group-item-text

- for text within .list -group-item, often in the paragraph tag or just in a span

Bootstrap Tables' Classes

.table

- for an ordinary table
- table heading has only border -bottom and <td>s have only border-top

.table-striped

- every odd line will be darker for a nicer look (using :nth-child selector)
- table heading <th> has only border -bottom and <td>s have only border-top
- should be combined with .table class

.table-bordered

- table with borders all around
- should be combined with .table class

.table-hover

- class for a table
- adds hover/focus effect on the table's tbody rows
- should be combined with .table class

.table-condensed

- should be combined with .table class
- for an ordinary table with half cell padding, so rows are not that high

.table-responsive

- class for a wrapper (! mostly div) of a table to keep it responsive in the way that scrollbar appears on extra small devices
- classes for either <tr> or just <td>
- sets background color for all <td> in a table row (or just one <td>) to grey (active) / green (success) / blue (info) / yellow (warning) / red (danger)

Bootstrap Forms' Classes

- don't forget that even form elements can be in the Bootstrap grid, so don't be afraid to use .row inside a form and then col-*-* wrappers

.form-group

- when creating vertical forms, it's a class for its wrapper (mostly div) of each label and its input, adds some margin-bottom

.form-control

- for every form element (input, textarea...), gives it 100% width
- to make it narrower, use a narrower wrapper (row and col-*-* divs)

.form-control-static

- for a paragraph that replaces a form element (input...)

.form-horizontal

- for horizontal forms where label is on the same line as input
- you have to use column classes (like col-xs-10) for each label and its input (or whateveer it is)
- still combined with .form-group wrapper and .form-control classes

.form-inline

- for a single line form (but still vertical on extra small screens)
- often used with the placeholder attribute to not need labels
- labels of inline forms should get .sr-only class to be positioned so that they can't be seen (but they won't get display: none or visibility: hidden property)

.input-lg

.input-sm

- makes a form elem ent like input or select either bigger or smaller than default

.input-group-lg
.input-group-sm

- classes for .input-group (see further)
- always used together with .input-group class

.input-group

- class for a wrapper of .input-group-addon or .input-group-btn
- used instead of form-group when the classes above are used

.input-group-addon

- class for a span that serves as a wrapper for either .glyphicon or a radio/checkbox input
- adds your choice (glyphicon...) on the inner edge of an input field with grey background
- the span with this class can be either before or after input, depends where you want the icon to show (the left or right side of the input)
- must be within .input-group

.input-group-btn

- class for a span that serves as a wrapper for a button that looks like it was within a form input
- the span with this class can be either before or after input, depends where you want the button to show (left or right side of the input)
- contains <button> (the button can also include .dropdown-toggle class, span with .caret class and data-toggle="dropdown" attribute for even more complicated buttons)
- must be within .input-group

.caret

- class for an empty span that creates a dropdown arrow

.help-block

- class for a span that serves as a help text for users
- has to be placed right after an input of a form

.has-success

.has-warning

.has-error

- classes for wrapping divs with .form-group class
- colours the label, input and .help -block span inside to green/yellow/red
- the label within the wrapper with one of these classes has to have .control-label class to be coloured as well

.has-feedback

- class for the wrapping divs with .form -group class when you also use glyphicon for the input within the wrapper

To disable input, use disabled="disabled" or readonly="readonly",

Bootstrap Image Classes

- images should have .img -responsive or .thumbnail to be responsive and be wrapped in a column wrapper (like col-xs-3...)

.img-rounded

- img class that provides little border-radius

.img-circle

- img class that provides 50% border-radius making the img circle shaped

.img-thumbnail

- img class that provides a 1px solid grey border, 4px padding, white background, display: inline-block property and fast CSS3 transition

.thumbnail

- class for a wrapping div (or <a> in case of gallery) of an image, adds some styles (border, padding, background, transition) and set's the image max-width: 100%, height: auto; display: block; and margin-left and margin-right: auto

.img-responsive

- img class that makes any image responsive (scaling to the parent element) as it has max-width: 100% and height: auto
- may not work if placed in a floating element!

.caption

- class for a div inside .thumbnail element (a or div) that may include h3, p, a (also buttons) for a proper margin/padding

.media

- class for a div (or li, whatever) that may contain an image, a heading, some <small> text and paragraphs
- ensures proper image margin/padding
- heading inside should also have media-heading attribute and should be (together with paragraphs) wrapped in a .media-body wrapper
- any image within should have .pull-left or .pull-right class to float

.media-list

- class for a list (ul, ol) that contains .media inside
- ensures better margin/padding

.glyphicon

- base class for glyphicons (font-based icons), sets font, its styles and positioning
- should be used on a span
- sets picture via font and :before CSS3 selector with content of the font
- must be used together with .glyphicon class to show proper image
- should be used on a span

Bootstrap Embed Classes

.embed-responsive

- general class for responsive embedding wrapper

.embed-responsive-4by3

.embed-responsive-16by9

- specific class for the embed wrapper with ratio already set
- used together with .embed-responsive class

.embed-responsive-item

- used on the iframe or whatever it is inside the wrapper

Bootstrap Navigation Classes

.nav

- the basic class for every navigation structure (<ul>), removes any list-style, left padding and bottom margin
- also sets display: block and position relative for its <li> children
- must be set on every navigation

.active

- class for a list item (nav item, li) that makes it look active / current

.nav-tabs

- class for navigation structure that gives it a tab style (proper borders + float: left)

.nav-pills

- class for navigation structure that gives it a pills style (background on active nav item)

.nav-stacked

- makes vertical pills navigation

- used together with .nav-pills (and .nav of course)

.dropdown

- class for an
- nav item to make it dropdown
- needs additional classes elsewhere to work (see below)

.dropdown-toggle

- class for an <a> which is within <li class="dropdown"> to make a dropdown menu
- needs additional classes elsewhere to work (see below)

.dropdown-menu

- class for an <ul> which is within <a class="dropdown-toggle"> to make a dropdown menu
- provides proper displaying and hiding
- the most important dropdown menu class
- needs .dropdown and .dropdown-toggle classes on parent elements (see above)

.nav-justified

- class for an <ul>, together with .nav-tabs or .nav-pills creates a navigation with centered text and links' parent elements of the same width

.disabled

- class for <li> in a navigation
- disables the link inside VISUALLY and also adds a special mouse cursor, but the link remains active - you need to remove the href attribute of the <a> to completely disable the link

Bootstrap Navbar Classes

- the difference between nav and navbar classes is the fact that navbar can collapse the whole nav (the nav can't do it on its own)

.navbar

- basic class for a navigation wrapper (usually <nav>), sets border, bottom margin, min-height and position: relative
- if it's not supposed to be fixed, can be placed anywhere in .container or .container-fluid

.navbar-default

- sets other basic styles for a navigation wrapper (background gradient and box-shadow) and also its items (including .active class on s)
- for dark colours, see .navbar-inverse below

.navbar-inverse

- variation of navbar-default but with dark colours
- never use navbar-default and navbar-inverse on one element as it doesn't make any sense

.navbar-brand

- sets styles for special hyperlinks and other "simple text" in a menu (float: left, bigger font size, some padding, text-shadow...)

.navbar-fixed-top

.navbar-fixed-bottom

- additional class to .navbar that makes the navigation fixed at the top (or bottom) of the page as you scroll
- used together with .navbar and also .navbar default (just to keep the nice look)
- you should add some top or bottom padding to your <body> to not let the navbar overflow your content

.navbar-static-top

- removes top, right and left border and also border-radius of your navigation

.breadcrumb

- class for <ul> that serves as breadcrumbs
- gives breadcrumbs a cool look

.pagination

- class for <ul> that serves as pagination
- properly styles <li> within this pagination also with .disabled and .active classes (keep in mind that .disabled doesn't remove link's functionality)

.pagination-lg
.pagination-sm

- used on <ul> together with .pagination, makes pagination either bigger or smaller than default

.pager

- used on <ul>, styles "Previous" and "Next" links to look better with a grey border
- its list items can have .previous, .next and also disabled classes for even better styling

.next

.previous

- used on <li> within <ul class="pager">, adds arrows to the links

Bootstrap 4: What's new in Bootstrap 4

New Classes in Bootstrap 4: Bootstrap made some notable changes while migrating from bootstrap 3 to bootstrap 4. Some of the major changes are listed below:

Forms:
-Rewrote both custom and default checkboxes and radios. Now.
-dropped .input-group-addon and .input-group-btn for two new classes, .input-group-prepend and .input-group-append

Grid System:
- xs grid classes have been modified to not require, So instead of .col-xs-6, it's now .col-6.

Images
-Renamed .img-responsive to .img-fluid.
-Renamed .img-rounded to .rounded
-Renamed .img-circle to .rounded-circle

Tables
-Responsive tables not require.just put the .table-responsive right on the <table>.
-Added a new .table-inverse option.
-Renamed contextual classes to have a .table--prefix. Hence .active, .success, .warning, .danger and .info to .table-active, .table-success, .table-warning, .table-danger and .table-info.

Forms
-Renamed .control-label to .col-form-label.
-Renamed .input-lg and .input-sm to .form-control-lg and .form-control-sm, respectively.

-Dropped .radio-inline and .checkbox-inline.
-Replaced .has-error, .has-warning, and .has-success classes with HTML5 form validation via CSS‘s :invalid and :valid pseudo-classes.

Buttons
-Renamed .btn-default to .btn-secondary.
-Change buttons‘ [disabled] to :disabled
-Removed explicit spacing between button groups in button toolbars

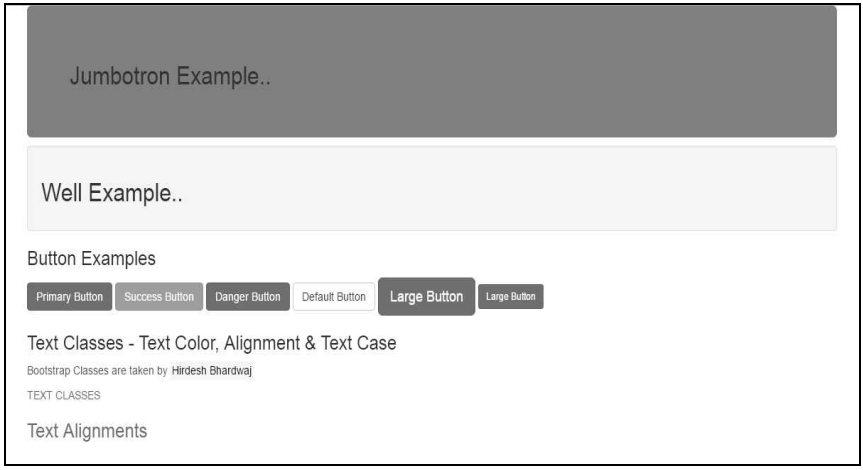
Panels
-.panel to .card, now built with flexbox.
-.panel-default removed and no replacement.
-.panel-group removed and no replacement.
-.panel-heading to .card-header
-.panel-title to .card-title.
-.panel-body to .card-body
-.panel-footer to .card-footer
-.panel-primary, .panel-success, .panel-info, .panel-warning, and .panel-danger have been dropped for .bg-, .text-, and .border utilities generated from our $theme-colors Sass map.

Example – Using Container, Jumbotron, well ,Button, Text Classes

```
<!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 101 Template</title>
<link href="css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container">
<div class="jumbotron">
<h2>Jumbotron Example..</h2>
</div>
<div class="well">
<h2>Well Example..</h2>
</div>
<h3>Button Examples</h3>
<button class="btn btn-primary">Primary Button</button>
<button class="btn btn-success">Success Button</button>
<button class="btn btn-danger">Danger Button</button>
<button class="btn btn-default">Default Button</button>
<button class="btn btn-primary btn-lg">Large Button</button>
```

```
<button class="btn btn-primary btn-sm">Large Button</button>
<br>
<h3>Text Classes - Text Color, Alignment & Text Case</h3>

<p class="text-danger">Bootstrap Classes are taken by
<mark>Hirdesh Bhardwaj</mark> </p>
<p class="text-primary text-uppercase">Text Classes</p>
<h3 class="text-warning text-left">Text Alignments</h3>

</div>
</body>
</html>
```

Output:



Example: Image Classes in Bootstrap


```
<!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 101 Template</title>
<link href="css/bootstrap.min.css" rel="stylesheet">  <!—Link to bootstrap CSS File -->
</head>
<body>
<img src=”myimage.jpg” class=”img-responsive”>
<img src=”myimage.jpg” class=”img-responsive img-thumbnail”>
<img src=”myimage.jpg” class=”img-responsive img-circle”>
<img src=”myimage.jpg” class=”img-responsive img-rounded”>
<img src="myimage.jpg" class="img-responsive img-circle img-thumbnail">
</body>
</html>
```

Example: Understanding Grid System –Rows & Cols

Bootstrap‘s grid system allows up to 12 columns across the page so we can create maximum 12 columns in a row. Example-

Let‘s create 3 equal columns having 4 each grid in a row

Let‘s create another row having 2columns of 8 and 4.

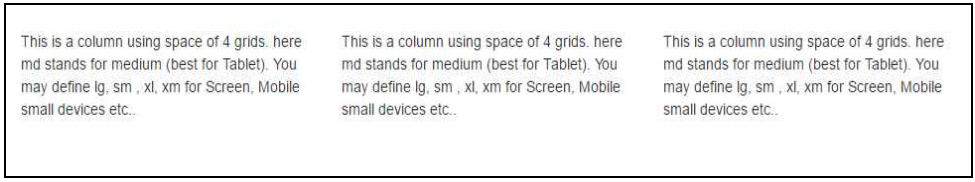
Example:

```
<!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 101 Template</title>
```

```
<link href="css/bootstrap.min.css" rel="stylesheet"> <! —Link to bootstrap CSS File -->

</head>

<body>

<div class="container">

<div class="row">

        <div class="col-md-4">

This is a column using space of 4 grids. here md stands for medium (best for Tablet). You may define lg, sm , xl, xm for Screen, Mobile small devices etc..

      </div>

        <div class="col-md-4">

This is a column using space of  4 grids. here md stands for medium (best for Tablet). You may define lg, sm , xl, xm for Screen, Mobile small devices etc..

       </div>

        <div class="col-md-4">

This is a column using space of 4 grids. here md stands for medium (best for Tablet). You may define lg, sm , xl, xm for Screen, Mobile small devices etc..
```

```
    </div>
</div> <!-- row end -->
</div>
</body>
</html>
```

Output:



Panel in Bootstrap

A panel is a bordered and stylish box having some padding around its content and sections like heading, body and footer.

Code Example:

```
<!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 101 Template</title>

<link href="css/bootstrap.min.css" rel="stylesheet"> <! —Link to bootstrap CSS File -->

</head>

<body>

<div class="container">

<div class="row"> <!—row start -->

<div class="col-md-4">

<div class="panel panel-primary"> <!—panel in blue border -->

<div class="panel-heading">About Panel</div> <! —heading tag not required here -->

<div class="panel-body"> <!—panel main content area -->

A panel is a bordered and stylish box having some padding around its content and sections like heading, body and footer.

</div>

<div class="panel-footer">Learn More...</div>
```

```
</div>

</div>

<div class="col-md-4">

<div class="panel panel-danger">

<div class="panel-heading">About Panel</div>

<div class="panel-body">

A panel is a bordered and stylish bo x having some padding around its content and sections like heading, body and footer.

</div>

<div class="panel-footer">Learn More...</div>

</div>

</div>

<div class="col-md-4">

<div class="panel panel-success">

<div class="panel-heading">About Panel</div>

<div class="panel-body">

A panel is a bordered and stylish box having some padding around its content and sections like heading, body and footer.
```

```
</div>

<div class="panel-footer">Learn More...</div>

</div>

</div>

</div> <!-- row end -->

</div>

</body>

</html>
```

Output:

Table Example in Bootstrap

Some table classes used to create table in bootstrap are table, table-responsive, table-bordered, table-striped, table-hover etc...

table-responsive

Is used to create a responsive table. The table will then scroll horizontally on small devices.

table-striped

Is used to adds zebra-stripes to a table.

table-bordered

Is used to adds borders on the table.

table-hover

Is used to enables a hover state on table rows i.e. a gray background color will automatically on mouse over.

table- condensed
Is used to makes a table more compact by reducing padding in half.

Example

```
<!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 101 Template</title>
```

```
<link href="css/bootstrap.min.css" rel="stylesheet"> <! —Link to boostrap CSS File -->

</head>

<body>

<div class="container">

<table class="table table-responsive table-striped table-bordered">

<tr>

<th>Name</th>

<th>City</th>

<th>Dept.</th>

</tr>

<tr>

<td>Rehaan</td>

<td>Noida</td>

<th>IT</th>

</tr>

<tr>

<td>Pooja Dayal</td>

<td>Udaipur</td>

<th>Development</th>

</tr>
```

```
<tr>
<td>Sonali Jain</td>
<td>Agra</td>
<th>Sales</th>
</tr>
</table>
</div>
</body>
</html>
```

Output

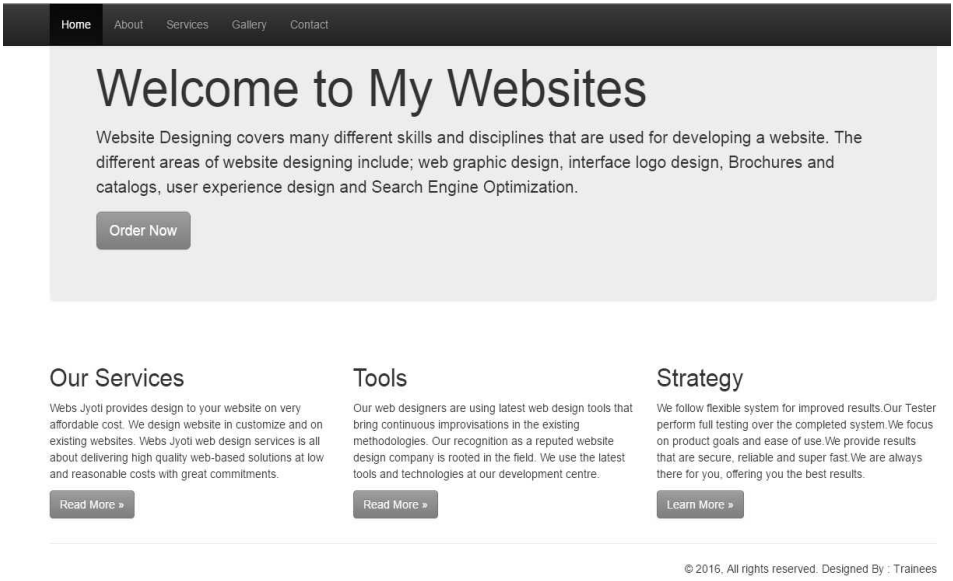
Name	City	Dept.
Rehaan	Noida	IT
Pooja Dayal	Udaipur	Development
Sonali Jain	Agra	Sales

Nav-bar Example

```
<nav class="navbar navbar-default">
 <div class="container-fluid">
 <div class="navbar-header">
 <a class="navbar-brand" href="#">Webs Jyoti</a>
```

```
  </div>
   <ul class="nav navbar-nav">
     <li><a href="#">Home</a></li>
     <li><a href="#">About us</a></li>
      <li class="active"><a href="#">Gallery</a></li>
     <li><a href="#">Contact</a></li>
    </ul>
 </div>
</nav>
</div>
</body>
</html>
```

Sample Design using Bootstrap



Code

```
<!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 Template</title>
```

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

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

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

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

</head>

<nav id="myNavbar" class="navbar navbar-default navbar-inverse
navbar-fixed-top" role="navigation">

<!-- Code for mobile display -->

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#navbarCollapse">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>
```

```
<!-- main menu items -->

<div class="collapse navbar-collapse" id="navbarCollapse">

<ul class="nav navbar-nav">

<li class="active"><a href="#" >Home</a></li>

<li><a href="#" >About</a></li>

<li><a href="#" >Services</a></li>

<li><a href="#" >Gallery</a></li>

<li><a href="#" >Contact</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="jumbotron">

<h1>Welcome to My Websites</h1>

<p>Website Designing covers many different skills and disciplines that are used for developing a website. The different areas of website designing include; web graphic design, interface logo
```

```
design, Brochures and catalogs, user experience design and SEO.</p>

<p><a href="#" target="_blank" class="btn btn-success btn-lg">Order Now</a></p>

</div>

<div class="row">

<div class="col-md-4">

<h2>Our Services</h2>

<p><strong>Webs Jyoti </strong> provides design to your website on very affordable cost. We design we  bsite in customize and on existing websites. Webs Jyoti web design services is all about delivering high quality web -based solutions at low and reasonable costs with great commitments.</p>

<p><a href="#" class="btn btn-success">Read More &amp;raquo;</a></p>

</div>

<div class="col-md-4">

<h2>Tools</h2>

<p>Our web designers are using latest web design tools that bring continuous improvisations in the existing methodologies. Our recognition as a reputed website design company is rooted in the
```

field. We use the lat est tools and technologies at our development centre.

Read More »

<h2>Strategy</h2>

We follow flexible system for improved results.Our Tester perform full testing ove r the completed system.We focus on product goals and ease of use.We provide results that are secure, reliable and super fast.We are always there for you, offering you the best results.

Learn More »

<footer>

© 2016, All rights reserved. Designed By : Trainees

```
</footer>

</div>

</div>

</div>

</body>

</html>
```

Thank You

I would like to say thank you p ublisher‘s team, who continued to encourage me throughout the entire publishing process. Thank you to everyone who has reviewed my book. My readers who taken time out of their day to explore it.

To get latest updates and details about my upcoming tutoria l and Books, you can subscribe me at www.websjyoti.com

To invite me to your campus or office for day long training programs write to us info@websjyoti.com

*****

About Author

Hirdesh Bhardwaj, born on August 15, 1987, at Aligarh, India. He is well-known for his best website/ software scripting and has nominated for the Oracle magazine "TOP IT Youngsters of India" in 2008.

In 2005, He completed his school education and moved to Gurgaon, Haryana for the higher education. He completed Diploma in Software Engineering from NIIT, Gurgaon in 2008.and Msc.IT from Karnataka University in 2010.

Hirdesh Bhardwaj got interested in writing while a freshman in high school. He wrote a Hindi poem "21st Century", later published in June 2009.

In 2009 he joined The Ardee Group, as a developer and IT head. Further he build Library Management System, Student management System, MLM System, Online Survey and many Applications using PHP/ Mysql.

Besides books, Hirdesh has about 250 scripts and tutorials will be publishing in upcoming years.

In 2009 He founded "Webs Jyoti"- Development & Training Organization. Webs Jyoti has developed about 350+ Websites and web applications.

You may reach the author at:
info@websjyoti.com

Also available as ebook
₹ 230/-
ACADEMIC
ISBN 978-93-90655-24-3
9 789390 655243 >

BOOKS CLINIC