



WEB APPLICATIONS DEVELOPMENT WITH PHP & MYSQL

Contents

Web Sites, Web Pages, Web Servers	4
Web Hosting.....	4
Some good free hosting:.....	4
Web Technologies.....	4
Using Bootstrap.....	5
Integrating Bootstrap to your Project.....	5
Popular Bootstrap Components	5
Navigation	5
Carousel	6
Grids	6
Table.....	7
Images	7
Alerts	8
Buttons.....	9
Button groups	9
Dropdown	9
Collapse.....	10
Modal	10
PHP Programming.....	11
Understanding Web Design and Web Development.....	11
Installation of the tools for development.....	11
Review of HTML input forms	11
PHP Coding.....	12
Opening and closing a php code block	12
Displaying texts and statement.....	12
Commenting.....	12

Declaring variables	13
Data types	13
Operators	13
Getting User Input from The Html Form.....	13
Conditional Statements	13
Sending and Processing Input and Output on the Same Page.....	15
Loop Structures	16
Arrays	17
Creating and Calling Functions Within the Program.....	18
The include() And require() Functions	18
Common String Functions.....	19
Date and Time Functions	19
PHP Cookies	20
PHP Sessions	20
PHP FILE HANDLING	21
PHP File Copy, Move And Delete	21
PHP and MYSQL Database Connectivity:	22
Crash course on databases and database objects (table, queries, forms, and reports)	22
Crash course on SQL statements	22
Setting up a MySQL server and creating a sample database.....	22
Connecting, querying, retrieving and displaying records.	22
Inserting, updating, deleting and searching records	23
Sample script for displaying a database table with edit / delete links each row	24
More Topics	25
Opening your locally hosted site for access.....	25
Uploading your site to a free web hosting site (ex. 000webhost.com)	26
Uploading files to your server.....	26

Downloading files from your server.....	26
Appendix A: Miscellaneous Useful PHP Functions.....	27
Converting input to HTML Special Characters	27
Form validation and Input Sanitation	27
Scan and list contents of a directory.....	27

Web Sites, Web Pages, Web Servers

- A web page is a collection of information that usually represent a part of a larger set or body.
- A web site is a collection of one or more web pages
- A web server is a computer (the host) for a web site.

Web Hosting

- Local hosting for local access
- Local hosting for intranet access
- Local hosting for public access
- Remote hosting

Some good free hosting:

- ✓ Hostinger.ph, 000webhost.com, Wix.com, Bluehost.com, Freewebhostingarea.com

Web Technologies

- a. For web structure:
Html, Xhtml
- b. For web designs and layout:
 - ✓ Cascading Style Sheet (CSS)
 - ✓ Bootstrap (also uses CSS)
 - ✓ Image editing applications like Adobe PhotoShop and the likes.
- c. For server-side scripts
 - ✓ PHP
 - ✓ ASP and ASP.NET
 - ✓ Python
 - ✓ Ruby
 - ✓ JavaScript (using Node.JS and Express)
- d. For client-side scripts
 - ✓ JavaScript (plain and vanilla)
 - ✓ Most JavaScript frameworks can also be used (as both server and client-side scripting) like ReactJS and AngularJS

Using Bootstrap

Integrating Bootstrap to your Project

```
<!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>My bootstrap test</title>

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

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

</head><body>

</body></html>
```

Popular Bootstrap Components

Navigation

(bs 3.3.7)

```
<div class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Home</a></li>
      <li class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">Page 1
        <span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="#">Page 1-1</a></li>
          <li><a href="#">Page 1-2</a></li>
          <li><a href="#">Page 1-3</a></li>
        </ul>
      </li>
      <li><a href="#">Page 2</a></li>
      <li><a href="#">Page 3</a></li>
```

```
</ul>
</div>
</div>
```

Carousel

```
<!--slideshow-->
<div id="myCarousel" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>

  <!-- Wrapper for slides -->
  <div class="carousel-inner">
    <div class="item active">
      
    </div>

    <div class="item">
      
    </div>

    <div class="item">
      
    </div>
  </div>

  <!-- Left and right controls -->
  <a class="left carousel-control" href="#myCarousel" data-slide="prev">
    <span class="glyphicon glyphicon-chevron-left"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="right carousel-control" href="#myCarousel" data-slide="next">
    <span class="glyphicon glyphicon-chevron-right"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

Grids

Bootstrap have 12 columns horizontally. Create a div for the row then distribute the number of columns to each div inside the row.

[3 equal columns]

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
```

```
<div class="col-sm-4">.col-sm-4</div>
</div>
```

[unequal columns]

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-8">.col-sm-8</div>
</div>
```

Table

[table hover, striped and responsive]

```
<table class="table table-hover table-responsive table-striped">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>Doe</td>
      <td>john@example.com</td>
    </tr>
    <tr>
      <td>Mary</td>
      <td>Moe</td>
      <td>mary@example.com</td>
    </tr>
    <tr>
      <td>July</td>
      <td>Dooley</td>
      <td>july@example.com</td>
    </tr>
  </tbody>
</table>
```

Images

[rounded corners]

```

```

[circular]

```

```

[thumbnail]

```

```


[linked thumbnails with caption and grid]

```
<div class="row">
  <div class="col-md-4">
    <div class="thumbnail">
      <a href="/w3images/lights.jpg">
        
        <div class="caption">
          <p>Lorem ipsum...</p>
        </div>
      </a>
    </div>
  </div>
  <div class="col-md-4">
    <div class="thumbnail">
      <a href="/w3images/nature.jpg">
        
        <div class="caption">
          <p>Lorem ipsum...</p>
        </div>
      </a>
    </div>
  </div>
  <div class="col-md-4">
    <div class="thumbnail">
      <a href="/w3images/fjords.jpg">
        
        <div class="caption">
          <p>Lorem ipsum...</p>
        </div>
      </a>
    </div>
  </div>
</div>
```

Alerts

```
<div class="alert alert-success">
  <strong>Success!</strong> Indicates a successful or positive action.
</div>

<div class="alert alert-info">
```

```
<strong>Info!</strong> Indicates a neutral informative change or action.
</div>

<div class="alert alert-warning">
  <strong>Warning!</strong> Indicates a warning that might need attention.
</div>

<div class="alert alert-danger">
  <strong>Danger!</strong> Indicates a dangerous or potentially negative action.
</div>
```

Buttons

```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
```

Button groups

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <button type="button" class="btn btn-primary">Sony</button>
</div>
```

Dropdown

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" data-toggle="dropdown">Dropdown Example
  <span class="caret"></span>
</button>
<ul class="dropdown-menu">
  <li><a href="#">HTML</a></li>
  <li><a href="#">CSS</a></li>
  <li><a href="#">JavaScript</a></li>
  <li class="divider"></li>
  <li><a href="#">others</a></li>
</ul>
```

```
</div>
```

Collapse

```
<button data-toggle="collapse" data-target="#demo">Collapsible</button>
```

```
<div id="demo" class="collapse">
```

```
  Lorem ipsum dolor text....
```

```
</div>
```

Modal

```
<!-- Trigger the modal with a button -->
```

```
<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal">Open  
Modal</button>
```

```
<!-- Modal -->
```

```
<div id="myModal" class="modal fade" role="dialog">
```

```
  <div class="modal-dialog">
```

```
    <!-- Modal content-->
```

```
    <div class="modal-content">
```

```
      <div class="modal-header">
```

```
        <button type="button" class="close" data-dismiss="modal">&times;</button>
```

```
        <h4 class="modal-title">Modal Header</h4>
```

```
      </div>
```

```
      <div class="modal-body">
```

```
        <p>Some text in the modal.</p>
```

```
      </div>
```

```
      <div class="modal-footer">
```

```
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
```

```
      </div>
```

```
    </div>
```

```
  </div>
```

```
</div>
```

PHP Programming

PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "PHP: Hypertext Preprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications that can publish a blog, remotely control hardware, or run a powerful website. Of course, to accomplish something such as this, you need a database application such as MySQL.

PHP is a server-side scripting language designed primarily for web development but is also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

Understanding Web Design and Web Development

Web designing handles:

- a. Structuring and backbone (html / xhtml)
- b. Design and layout (css)
- c. Media content for appearance (image / animation / video content)

Web development handles:

- a. Web GUI events
- b. Web page behavior
- c. Interpreting data-driven request and responses
- d. File handling and manipulation
- e. Utilization and management of web site files and resources

Installation of the tools for development

- a. Apache, MYSQL, PHP (WAMP, LAMP, XAMP, individual installation)
- b. Sublime / Notepad++

Review of HTML input forms

```
<form method="post/get" action="" *enctype="multipart/form-data" onSubmit="?">  
  
<input type="text"  
  name="?"  
  pattern="[0-9a-zA-Z]{3,8}"  
  size="20"  
  maxlength="18"  
  required
```

readonly>

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

<input type="email" name="?">

<input type="radio" name="?" value="?" checked>

<input type="checkbox" name="?" value="?" checked>

<input type="button" name="?" value="?" onclick="?">

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

<input type="reset" name="?" value="?">

<textarea name="?" rows="?" cols="?"></textarea>

<select name="?" *size="4">

<optgroup label="?">

<option value="?" selected></option>

<option value="?"></option>

<option value="?"></option>

</optgroup>

</select>

<input type="file" name="?" accept="images/*">

<input type="hidden" name="?" value="?">

<input type="date" name="?">

PHP Coding

Opening and closing a php code block

```
<?php
...Php code goes here...
?>
```

Displaying texts and statement

```
echo "this is a sample php statement";
```

Commenting

```
//this is a single-line comment
/*This is a
Multi-line
Comment*/
```

Declaring variables

```
//unlike other programming languages, php automatically selects the data type of a variable

//declaring variables
$name = 'janaa';
$address = 'valenzuela city';
$birthyear = 2012;

//inserting variables in a php statement within double quotes
echo "hi, my name is <b>$name</b>, I live at <b>$address</b>. I was born on the year
<b>$birthyear</b>."

//concatenating (symbol=".") variables and string constant
echo "hi ".$name."you live at ".$address." and was born on ".$birthyear.".";

//it is also possible to assign a value to multiple variables simultaneously
$var1=$var2=$var3=60;
```

Data types

Boolean (true/false), Integer, Floating point (with decimal places), String

Operators

Arithmetic (+, -, *, /, %)
Comparison (>, <, ==, >=, <=, !=)
Logical (and[&&], or[||])

Getting User Input from The Html Form

```
//this usually involves two pages:
//page for the input form→

<html><body>
<form action="message.php" method="post">
enter your message: <input type="text" name="message" size="30">
<input type="submit" name="submit1" value="Send">
</form></body></html>
```

```
//and a page that processes the action
//this will, for example, be "message.php"
<?php

    //retrieve the other page's data from the input form and put it in a variable
    $inputmessage = $_POST['message'];
    //then use it
    echo "you said: <i>$inputmessage</i>
?>
```

Conditional Statements

```
//the following shows a single-argument condition
if (condition)
```

```
{
...result1...;
...result2...;
}
```

//the following shows a two-argument condition

```
if (condition)
{
...result1...;
...result2...;
}
else
{
...result1...;
...result2...;
}
```

//the following shows a multi-argument condition

```
if(condition)
{
...result1...;
...result2...;
}
elseif(condition)
{
...result1...;
...result2...;
}
else
{
...result1...;
...result2...;
}
```

//and a nested if condition

```
if(condition)
{
    if(condition){result;}
    else{result;}
}else{
    Result;
}
```

//aside from the "if" structure, you can also use the "switch" statement

```
switch(variable to test)
{
case 'value1':    ...result1...;
                  ...result2...;
                  break;
case 'value2':    ...result1...;
                  ...result2...;
                  break;
case 'value3':    ...result1...;
```

```

        ...result2...;
        break;
default:
        ...result1...;
        break;
}

```

Sending and Processing Input and Output on the Same Page

```

//this involves testing the page if the submit button is already pressed
<html><body>

<?php    if (!isset($_POST['submit'])){    ?>

//breaks the php script to insert html form
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
Enter name: <input type="text" name="txt_name" size="30">
<input type="submit" name="btn_submit" value="send">
</form>

//then resumes with the php script execution
<?php    }else {
$age = $_POST['age'];
if ($age>=18)
{ echo "ready to vote";}
else
{ echo "not yet ready to vote";}
}?></body></html>

```

multicheckbox array example:

```

<html>
<title>multicheck form</title>
<body>

<form method="post" action="multicheckform.php">
check 1 <input type="checkbox" name="checkvar[]" value="a"><br>
check 2 <input type="checkbox" name="checkvar[]" value="b"><br>
check 3 <input type="checkbox" name="checkvar[]" value="c"><br>
check 4 <input type="checkbox" name="checkvar[]" value="d"><br>
check 5 <input type="checkbox" name="checkvar[]" value="e"><br>
check 6 <input type="checkbox" name="checkvar[]" value="f"><br>
check 7 <input type="checkbox" name="checkvar[]" value="g"><br>
check 8 <input type="checkbox" name="checkvar[]" value="h"><br>
<input type="submit" name="btn1" value="submit">
</form>

<br><br><br><br>
<?php
if (isset($_POST["btn1"])){
echo implode(", ", $_POST["checkvar"]);
}
?>

```



```
<br><br>
<a href="multicheckprocess.php">reload</a>
</body>
</html>
```

Loop Structures

```
while (condition)
{
...Action...;
...limiting counter (increment/decrement)...;
}

do
{
...action...;
...limiting counter (increment/decrement);
}while (condition)
```

```
for(var=initialvalue, condition, increment/decrement)
{...action...;}
```

Example: (looping for populating contents of a menu or list)

```
<select>

<?php
$curr_year=date['Y'];
for ($year_counter=1940;$year_counter<=$curr_year;$year_counter++)
{
echo "<option value="<?php echo $year_counter; ?>"> $year_counter </option>";
}
?>
</select>
```

Example: (loop to create a multiplication table)

```
<html><body>

<?php
If (!isset($_POST['submit1']))
{
?>

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
Enter number for the multiplication table <input type="text" name="text1" size="2">
<input type="submit" name="submit1" value="send">
</form>

<?php
}
else
{
```

```

$numbertomultiply=$_POST['text'];
for($counter=0;$counter<=#numbertomultiply;$counter++)
{
    $answer=$numbertomultiply*$counter;
    echo "the number $numbertomultiply when multiplied to $counter is $answer ";
}
?>

</body></html>

```

Arrays

Create and use arrays

```

<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>

```

Count array

```

<?php
$cars = array("Volvo", "BMW", "Toyota");
echo count($cars);
?>

```

Loop through an array

```

<?php
$cars = array("Volvo", "BMW", "Toyota");
$arlength = count($cars);

for($x = 0; $x < $arlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>

```

Associative arrays

```

//create
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

//or
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";

//use
echo "Peter is " . $age['Peter'] . " years old.";

//loop through an associative array
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
}

```

```
    echo "<br>";  
}
```

Creating and Calling Functions Within the Program

Sample1

```
<?php  
//functions are set of programs that can easily be called from anywhere in the program. Re-usable.  
  
function mysamplefunction()  
{  
    echo "hi, im a sample function.";  
}  
  
//to call a function  
mysamplefunction();  
?>
```

Sample2(functions with arguments and returning the answer of a function for a variable)

```
<?php  
function addvalues($num1,$num2)  
{  
    $num1=0;  
    $num2=0;  
    $mytotal=$num1+$num2;  
    return ($mytotal);  
}  
  
//assign the functions' return value to variable with the function's arguments  
$myanswer=addvalues(10,45);  
  
//give another variable a different answer using the same function  
$anotheranswer=addvalues(34,78);  
  
//or just output the function's answer on the page  
addvalues(23,945);  
?>
```

The include() And require() Functions

Note: these two functions are used to call or add a different module or page into your current page. **include()** continues the script even though you encounter problems while the **require()** function shows a fatal error then halts the script processing.

(myheader.php)

```
<?php  
echo "this will be the header of the page";  
echo "<br>";  
echo "<a href='index.php'>home</a> | ";  
echo "<a href='page1..php'>about</a> | ";  
echo "<a href='page2.php'>categories</a> | ";  
?>
```

(myfooter.php)

```
<?php
$curr_yr=date("Y");
echo "<center><a href='about.php'>yourcompany inc.</a><br>";
echo "all rights reserved 1990-$curr_yr</center>";
?>
```

(mainpage.php/index.php)

```
<?php
include('myheader.php');

//other codes here

include('myfooter.php');
?>
```

Common String Functions

- echo () //outputs strings
- trim () //strips whitespaces on left side of string
- strlen()//counts the number of characters
- substr("string",integer position,number of characters)//gets string within string, starts from right to left
- str_word_count () //counts number of words in a string
- strcasecmp () //compares two strings (case insensitive)
- strcmp () //compares to strings (case sensitive)
- strtolower () //converts strings to lower case
- strtoupper () //converts strings to upper case
- stripslashes() //removes backslashes
- htmlspecialchars() //converts special char into html format

Date and Time Functions

Ex:

```
$monthnameshort=date("M"); //JAN
$monthnamefull=date("F"); //JANUARY
$monthnum=date("m"); //1-12
$dayname=date("D"); //SUN
$daynamefull=date("l"); //(lowercase "L") SUNDAY
$daynum=date("d"); //1-31
$yearfull=date("Y"); //1999
$yearshort=date("y"); //99
$hour12=date("h"); //1-12
$hour24=date("H"); //0-23
$minute=date("i"); //0-59
$seconds=date("s"); //0-59
$timeofday=date("a"); //am-pm
```

Note: affected by time zone, adjust time zone in php.ini.

PHP Cookies

//create a cookie, syntax→ setcookie(cookie name, value, expire)

```
<?php
setcookie("user","alex",time()+3600);
?>
```

//to use or view the cookie

```
<?php
echo $_COOKIE("user");
?>
```

//to view all cookies

```
<?php
print_r($_COOKIE);
?>
```

//try checking cookies with the isset() function too!

PHP Sessions

//start a session use session_start()

//MUST be called BEFORE the <html> tag

```
<?php
session_start();
```

```
$_SESSION['views']=1; //here is a sample session variable
?>
```

```
<html><body>
```

//then we can use or check the session variable

```
<?php
if(!isset($_SESSION['views']))
{
    $_SESSION['views']=1;
}
else
{
    $_SESSION['views']=$_SESSION['views']+1;
}
```

```
echo "pageviews: ".$_SESSION['views'];
?>
```

```
</body></html>
```

//to release a session variable use

```
<?php
unset($_SESSION['views']);
```

```
?>
```

```
//to destroy a session  
<?php  
session_destroy();  
?>
```

PHP FILE HANDLING

1. Opening and reading a file

```
<?php  
//set file to read  
$file='sample.txt' or die("file not found.");  
//set file handling and selecting open mode  
$fh=fopen($file,'r') or die("could not open file");  
//read the contents and put it in a variable  
$data=fread($fh,filesize($file)) or die("could not open the file");  
//close the connection  
fclose($fh);  
//output the data  
echo $data;  
?>
```

Note:

*fopen() parameters are as follows: fopen(filepath,'mode')

Modes --- 'r'(read only mode)

'w'(destroys existing contents and overwrites it)

'a'(opens and preserves file contents and appends)

2. Opening and writing to file

```
<?php  
//set file to read  
$file='sample.txt' or die("file not found.");  
//set file handling and selecting open mode  
//('r'=readonly,'w'=overwrite existing,'a'=preserve and append new contents)  
$fh=fopen($file,'a') or die("could not open file");  
//write to file  
fwrite($fh,"this appends a new line") or die("could not append to file");  
//close the file  
fclose($fh);  
?>
```

PHP File Copy, Move And Delete

*note: doesnt work with <input type=file>

1. to copy

```
copy(source,destination)
```

2. to move

```
rename(source,destination)
```

3. to delete file

```
unlink(file)
```

4. to create directory

```
mkdir(path and directory name)
```

5. to remove a directory

```
rmdir(path and directory name)
```

PHP and MYSQL Database Connectivity:

Crash course on databases and database objects (table, queries, forms, and reports)

(Trainer Discussion)

Crash course on SQL statements

Ex:

```
SELECT [field1],[field2],... FROM [tablename] WHERE [field] = '[value]' ORDER BY [field] asc/desc
```

```
SELECT [field1],[field2],... FROM [tablename] WHERE [field] like '[value]%' ORDER BY [field] asc/desc
```

//the use of LIKE gets value that may look like –or may contain words like—the value used.

```
INSERT INTO [tablename] (field1, field2, ...) VALUES ('value1', 'value2',...)
```

```
UPDATE [tablename] SET [field1]='[newvalue]', [field2]='[newvalue]'... WHERE [field]='[oldvalue]'
```

```
DELETE FROM [tablename] WHERE [field] = '[value]'
```

Setting up a MySQL server and creating a sample database.

(Trainer discussion)

Connecting, querying, retrieving and displaying records.

Note: be very careful when copy-pasting scripts/codes from a different file (ex. *.doc) because it may have differences in symbols and characters. One may not recognize the font face the other is using.

```
<html><body>
<?php
//set db server access variables
$host="localhost";
$user="root";
```

```

$pass="root";
$db="testdb";

//open connection
$con=mysqli_connect($host, $user, $pass, $db) or die("unable to connect.");

//create query
$query="select * from table1";

//execute query
$result=mysqli_query($con,$query) or die("error in query: $query".mysqli_error($con));

//test if the db have records
if(mysqli_num_rows($result)>0){

//if yes, print them one after another.

//I'll be using a looped table to contain the records
echo "<table cellpadding=10 border=1>";

//now the loop for the rows and columns
while($row=mysqli_fetch_row($result))
{
//the loop shows the description of the row and the $row array is for the columns.
echo "<tr>";
echo "<td>".$row[0]."</td>";
echo "<td>".$row[1]."</td>";
echo "<td>".$row[2]."</td>";
echo "</tr>";
}
echo "</table>";
}

else
{
echo "no records";
}

//close connection
mysqli_close($con);
?>
</body></html>

```

Inserting, updating, deleting and searching records

(Trainer Activity)

Sample script for displaying a database table with edit / delete links each row

NOTE:

1. each row will have an edit and delete link
2. links have confirmation
3. each row link (edit / delete) creates a custom url that the other page can grab using the "GET" method

view page

```
<html>
<title>view db</title>
<body>
<?php
$host="localhost";
$user="john";
$pass="123";
$db="db1";

$con=mysqli_connect($host,$user,$pass,$db) or die("error in connection");

$query="select * from table1 order by empid";
$result=mysqli_query($con,$query) or die(mysqli_error($con));

if(mysqli_num_rows($result)>0)
{
echo "<table width=40% align=center border=0>";
    while(($row=mysqli_fetch_row($result))!=null)
    {
        echo "<tr>";
        echo "<td>".$row[0]."</td>";
        echo "<td>".$row[1]."</td>";
        echo "<td>".$row[2]."</td>";
        echo "<td><a href=edit.php?empid=".$row[0].\" onClick=\"return confirm('sure?');\">edit</a></td>";
        echo "<td><a href=delete.php?empid=".$row[0].\" onClick=\"return confirm('sure?');\">delete</a></td>";
        echo "</tr>";
    }
echo "</table><br><center>records displayed:".mysqli_num_rows($result)."</center>";
}
else
{
echo "no records found";
}

mysqli_close($con);
?>
<br><br>
<a href="multicheckform.php">multi check form</a>
</body>
</html>
```

```
<html>
<title>edit page</title>
<body>

<?php
echo "you will be editing record with employee id: <b><u>".$_GET["empid"]."</u></b> ";
?>

<a href="index.php">back</a>

</body>
</html>
```

More Topics

Opening your locally hosted site for access

- a. Intranet and Internet access
 - On wamp controls, click “put online”
 - Apache settings → httpd.conf
 - a. Change servername [localhost:80] → [your pc’s static IP:80]
 - b. <Directory />
//Modify: Require all denied to
Require all granted
</Directory>
 - Apache settings → httpd-vhosts.conf

```
<VirtualHost *:80>
    ServerName 192.168.1.110
    ServerAlias localhost
    DocumentRoot c:/wamp/www
    <Directory "c:/wamp/www/">
        Options +Indexes +Includes +FollowSymLinks +MultiViews
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

- Windows firewall → advanced settings → inbound rules → new inbound rule for port 80 (apache)
- Port forward on your router
- Accessing the site:
 - a. Thru local intranet [<http://pcname:port> or <http://privateIP:port>]
 - b. Thru public internet [<http://yourpurchaseddomain.com> or <http://publicIP:port>]

Uploading your site to a free web hosting site (ex. 000webhost.com)

- a. Using php mail() functions with web host's mail provider

```
<?php
$to=S_POST['text1'];
$subject=$_POST['text2'];
$message=$_POST['area1'];
$headers="FROM:".$_POST['text3'];
$mailcheck=mail($to,$subject,$message,$headers);
if($mailcheck){ echo "success"; } else { echo "fail";}
?>
```

Uploading files to your server

Upload form:

```
<html><title>file upload</title><body>
file upload page<br><br>
<form method="post" action="upload_process.php" enctype="multipart/form-data">
<label for="file">Filename</label><br>
<input type="file" name="file"><br><br>
<input type="submit" name="submit" value="click to upload"><br>
</form></body></html>
```

Basic Upload Script:

```
<html><title>upload process</title><body>
<?php
if($_FILES["file"]["error"]>0){
echo "error ".$_FILES["file"]["error"]."<br>";
}else{
echo "upload: <b><i><u>".$_FILES["file"]["name"]."</u></i></b><br>";
echo "type: <b><i><u>".$_FILES["file"]["type"]."</u></i></b><br>";
echo "size: <b><i><u>".$_FILES["file"]["size"] / 1024). " kb</u></i></b><br>";
move_uploaded_file($_FILES["file"]["tmp_name"],"upload/".$_FILES["file"]["name"]);
echo "stored in: <b><i><u>". "upload/".$_FILES["file"]["name"]."</u></i></b><br>";
}
?>
</body></html>
```

Downloading files from your server

- Prepare a folder for your downloadable files in your web directory. Add some files.
- Create hyperlinks on your page to trigger file downloads:

To try and open the file on the browser before download:

```
<a href="file1.pdf"> download file1 </a>
```

To force download of file:

```
<a href="file1.pdf" download> download file1 </a>
```

Appendix A: Miscellaneous Useful PHP Functions

Converting input to HTML Special Characters

example:

```
$message = htmlspecialchars($_POST['text1']);
```

Form validation and Input Sanitation

```
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
```

Scan and list contents of a directory

```
<?php
$dir = "/images/";

// Sort in ascending order - this is default
$a = scandir($dir);

// Sort in descending order
$b = scandir($dir,1);

print_r($a);
print_r($b);
?>
```