



## Advanced Online Media

Dr. Cindy Royal

Texas State University - San Marcos

School of Journalism and Mass Communication

---

## Using PHP

### What is PHP?

- Server side, scripting language
- Designed for use with html
- Adds flexibility to html
- Syntax similar to programming languages like C, Java, Perl, and ASP

### PHP History

Started in 1994, based on C language to replace some Perl script CGI binary files, to make maintenance of Web site easier. Originally it was called Personal Home Page (PHP) Tools. Upgraded in 1998 by others (Zend) and given a new recursive name PHP Hypertext Preprocessor. Currently at Version 5 (5.4.13). Our Bluehost accounts have version 5.2.17. We are going to be working with these exercises directly on the server, but if you do more PHP work, you will want to install locally with something like MAMP (Mac Apache, MySQL and PHP). There are different installs for other platforms. See <http://www.mamp.info>.

### Why PHP?

- Open Source, Free Software
- Cross platform for development, deployment and use
- Powerful, robust, scalable
- Web Development Specific
- Can be object-oriented
- Excellent documentation at [www.php.net](http://www.php.net)
- Large, active development community (20 million Web sites)

PHP can be used by itself for some functionality (basic scripting, server-side includes), but can be very powerful with integrated with a database application like MySQL

### Simple PHP script

PHP is easily embedded in an html page. Simply open the php area with `<?php`, write your code, then close with `?>`. You can have multiple PHP areas in an html page. PHP files must have a .php extension.

```
<?php
```

```
echo "Hello World!";
```

```
?>
```

## Basic PHP Syntax

“echo” simply means to display it as html on the page. Text strings can be wrapped in “ or ‘. All statements are ended with a “;”.

PHP can also use variables. Variables must start with a \$. They are not wrapped in quotes.

```
<?php
```

```
$var1 = "Hello World!";  
echo $var1;
```

```
?>
```

And, you can concatenate variables with strings or other variables very easily. Just use a “.” to concatenate.

```
<?php
```

```
$var1 = "Cindy";  
  
echo "Hello " . $var1;
```

```
?>
```

Comments are helpful to use in programming to identify sections or to explain what you are doing. To comment a single line, use “//”. To comment a multi-line area, use “/\*” to open, and “\*/” to close.

You can also get into a lot of other detail with PHP, like if statements, loops, and functions. See the W3 Schools PHP tutorial at <http://www.w3schools.com/php/default.asp>. We’ll be looking at some of these items in existing scripts.

## Server-side includes

One good use of PHP is that of server-side includes. This lets you insert the contents of one html page into another. The benefit is that you can change the include page, thus changing it on every page that has it included.

You can achieve a similar effect by using frames or even iframes, but this is not recommended by developers. Frames are not as accessible, and the contents of frames must be entire html pages (head, body, etc.). An include page is a simple text file that includes the content you want to have included.

iFrame code:  
<iframe  
src ="file.html"  
width="100%">  
</iframe>

```
<?php  
include("filename.inc")
```

```
?>
```

The filename can have a php or txt extension as well, but the inc extension is useful in helping you identify and organize your files.

Server-side includes are great for footers or navigation or other sections of the page that need to be updated easily (like the Announcements section of our class site).

When PHP executes in the browser, you do not see the PHP code. You only see the html that it rendered. All processing is done on the server side.

### **Use PHP to Post to MySQL database**

Most of PHP's power is in how you can use it to post and retrieve data from a database. Content Management Systems like Wordpress and Drupal are based in PHP/MySQL.

### **5 steps of PHP/DB interaction**

1. Create a Connection to a DB
2. Select a DB
3. Perform a DB Query
4. Return User Data
5. Close Connection

First you must set up the MySQL database. You use can PHPMyAdmin to set up your fields. You have access to this with your Bluehost Account.

1. Go to your Bluehost account and login to the control panel - bluehost.com.
2. Under Databases, find MySQL Databases. Add a new database named "discussion". Then give it a username and password for a user, connect to the database you just created. Remember these items. Pay attention to how Bluehost names these items (with your Bluehost username preceding).
3. Now we will add fields to the database. Under Databases, find PHPMyAdmin. It's a great interface for working with your MySQL databases. You should see your "discussion" database. Open it.
4. Create a new table on it named "discuss1" with 4 columns. Then name the columns, Date, FirstName, LastName and Comments. Pay attention to capitalization. The Date, FirstName and Last Name will be set up as varchar with

a 35 character limit. Set up Comments as text (no character limit). Scroll down and Save.

Now, we will create php files to interact with the database. In PHP, you can connect to the database, then execute commands that will write form data to the database.

1. You'll be posting these files via the File Manager and then working with them via the File Manager's HTML Editor. Make a folder called php under your main domain in your public folder.
2. Create a form (use a text editor) that captures the necessary information (I have one for you to use called "discuss1.php"). Use the following action in the form field:  
<form name="frmName" method="post" action="d1.php">
3. You will need to include the file d1.php in the same folder as the form, or you will have to include a path.
4. Then, we create d1.php (I have it created for you). First, we put the necessary information in variables. You will need to adjust for your own database name, username and password. We create a connection variable with the mysql\_connect command, and then a mysql\_select\_db command.
5. Then, the mysql\_query function is used to create a variable that inserts our values for the variables into the database. Use the proper syntax as described below.
6. The header command redirects to the d1-reader.php page after it submits the query, then the mysql\_close is called.

### **d1.php**

```
<?php
```

```
$hostname= "localhost";  
$databasename="yourdatabasename";  
$username="yourusername";  
$password="yourpassword";  
$connection = mysql_connect($hostname,$username, $password);  
mysql_select_db($databasename, $connection);  
  
$_POST['Comments'] = nl2br($_POST['Comments']);  
//this allows the line breaks to occur in the reader  
  
$insertQuery = mysql_query("Insert into test (Date, FirstName, LastName, Comments)  
values  
('".$_POST['Date']."', '".$_POST['FirstName']."',  
".$_POST['LastName']."', '".$_POST['Comments']."'", $connection);  
  
header("Location:d1-reader.php");
```

```
mysql_close($connection);
?>
```

6. The reader page (d1-reader.php – you have it, just change the database and user info) is simple. It is a table, but the code is a loop that is used to create rows for the table as long as there is still data to read in the database.

### ***d1-reader.php***

```
<table border='1' cellspacing='0' cellpadding='5'
    width="750" style="border-collapse:collapse;
    font-family:sans-serif;">
<tr style="background-color:#B2AC79"><th>
Date</th><th>FirstName</th><th>LastName</th><th>Comments</th></tr>
```

```
<?PHP
$hostname= "localhost";
$databasename="yourdatabasename";
$username="yourusername";
$password="yourpassword";
$connection = mysql_connect($hostname,$username, $password);
mysql_select_db($databasename, $connection);
```

```
$result= @ mysql_query("SELECT * FROM test", $connection);
//selects projects for this user
```

```
while($row = mysql_fetch_array($result))
{
print("<tr><td>".$row["Date"]."</td><td>".$row["FirstName"]."</td><td>".$row["LastNam
e"]."</td><td>".$row["Comments"]."</td></tr>");
}
```

```
mysql_close($connection);
```

```
?>
```

```
</table>
```

Try it out to see if you can post data to the database and read it back in the browser.

Now, on your own, I want you to add a field for Twitter. Here are the things you will need to modify.

1. Add a field to the MySQL table (it can be varchar 35). Do this under Structure.
2. Add a row for Twitter to the table in discuss1.php that allows for input of Twitter name.
3. Modify d1.php to capture Twitter.
4. Modify d1-reader.php to display Twitter in head and body of table.
5. Test it out! Refresh the form. Try posting a new line to the database.
6. To troubleshoot, you can look at the database in PHPMysqlAdmin to see if your data got posted. If so, then you know the problem is in the reader.