

Chapter 1

1.1 What Is PHP?

This presentation is to answer the question, what is PHP?

PHP is one of the most popular scripting languages that are used for web development.

The cool thing about PHP is it's embedded inside of an HTML page itself. There are special script tags that you'll use to delineate PHP code from HTML code.

Its main purpose is to make web pages dynamic. Now, by dynamic, I don't mean cool (although they might be cool when we're done.) "Dynamic" means to make the web page display change based on something that the user inputs like their name, account number, or some data that is found in a database or file. So, dynamic means *changing*.

The nice thing about PHP is it's going to "live" on a web page. Those things about the web page, and about HTML code, that don't change we will just leave them in "regular" HTML code.

The PHP script will produce *new* HTML code, depending on user input and/or data in the database.

PHP is called a *server-side* language. It's designed to run on a computer server that host software to host web applications. It's not deployed locally on your computer. Nobody could get at it if it was.

The whole idea of Internet is to have other people go to your web pages that run PHP.

You develop PHP locally on our own computers using special software that I'm going to show you how to install later. It's really easy to learn and it is free. You can download it and deploy it, use it, and do anything you want with PHP.

You will use a special package that lets you get a web server, PHP, MySQL, and a few other tools. And fortunately, it's really easy to do. PHP code, in my opinion, is simpler to learn than other server-side languages. I'm familiar with other languages like Java Enterprise Edition, which run JSPs and Servlets, etc. and they are fairly complex. Also used is Microsoft's ASP which forces you to deal with various Microsoft features such as .NET. All of them work. All of them work well and do different things on the servers, but PHP is probably the easiest to learn and use.

To run PHP on a web server you must use special software. Runtime PHP software has to be installed on that server, otherwise, the PHP code will not be processed.

PHP is just code that can be used inside of an HTML page. The output from a PHP program itself is just another HTML page. You will soon see how that works.

PHP programs are called from one web page and return brand new web pages to the browser as a result. Look at the following diagram...



Figure 1.1.1 – How a PHP Program is used with a Browser

There is a little box representing the web page. It starts on your local browser. You click a button or a link and it calls a PHP program. It does its processing and then renders a brand new web page back to your browser.

Let's take a look at a real web page. This is a very simple one...

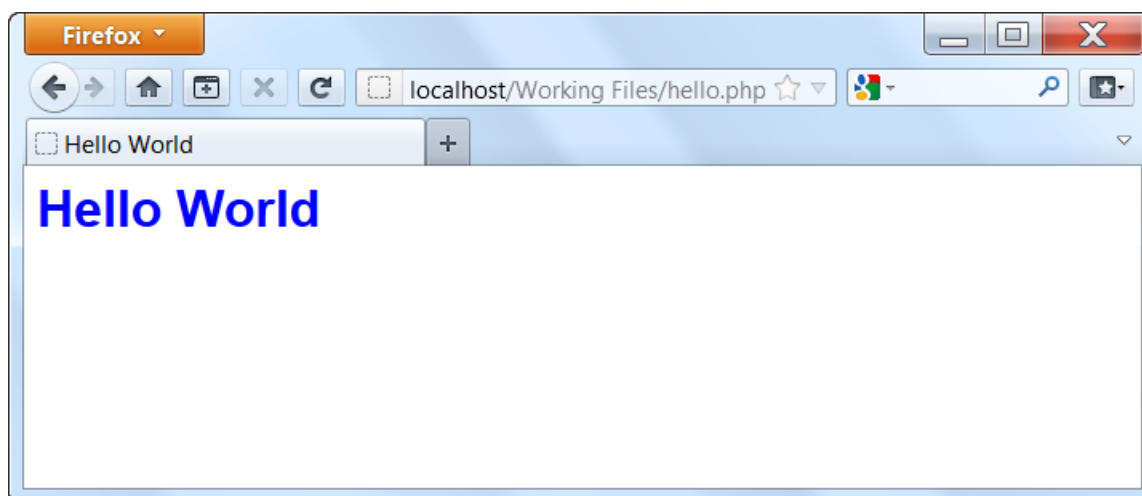


Figure 1.1.2 - Hello World web page display

The "Hello World" is blue. When I refresh the page, now it's red...

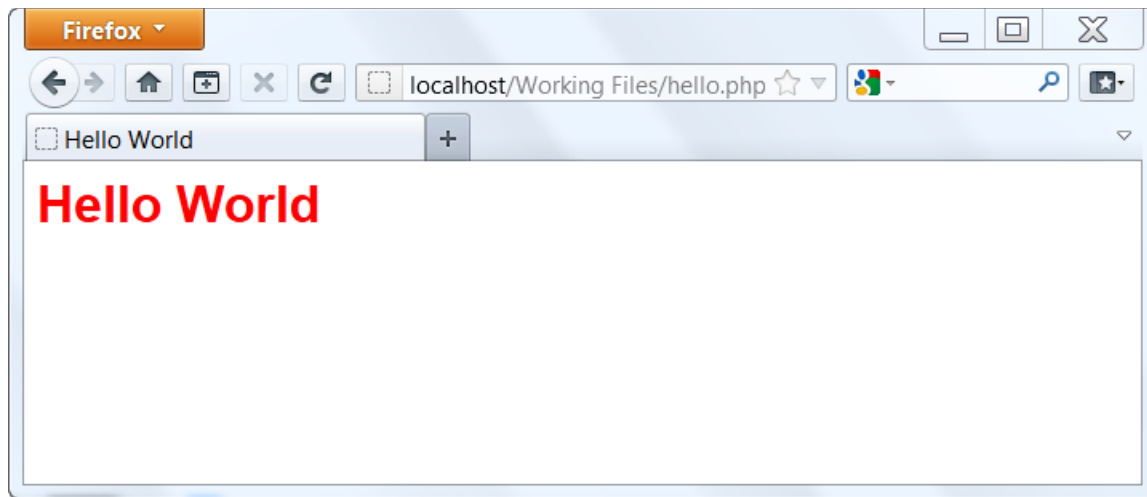


Figure 1.1.3 – Hello World web page display (2nd time)

Each time I refresh the page, the PHP program will pick a color from a list I specified in the PHP code.

Let's take a look. It is just a few lines of PHP code, don't worry if you don't understand them right now, you will later.

```
<body>
<?php

    $colorArray = array('blue', 'green', 'red', 'maroon', 'gray');

    $sec = date('s');    // Get the second on the clock

    $colorIndex = $sec % 5;    //Determines remainder

    print '<h1 style="color: '.$colorArray[$colorIndex].';">
        Hello World</h1>';    //Shows the heading on the page

?>
</body>
```

Figure 1.1.4 – Example PHP Program Code

This code simply...

1. Specifies a list of colors.
2. Determines what second (e.g. 1-60) it is on the computer's system clock when the user reloads the page.
3. Divides that number by 5 and saves the remainder. (Note: The remainder will always be 1, 2, 3, 4, or 5)
4. Uses the remainder to determine which color on the list to use for the color of the "Hello World" heading.

Again, you will learn about the specifics of these PHP statements later, as we proceed through the lessons.

1.2 What Is MySQL?

So what is MySQL?

MySQL is a version of SQL. We'll talk about that in a minute. People in the know don't say 'My' 'S' 'Q' 'L', they say 'My' 'Sequel'. And SQL is usually pronounced 'Sequel'.

SQL itself is a unified language specification that defines how to create database objects, query data in a database, modify it, control it, etc. within a relational database.

What is a relational database?

A relational database stores all the data inside database object called tables. It's a grid-like structure, very similar to an Excel spreadsheet.

MySQL is used for structuring, querying and changing information in these tables.

Table have rows and columns very much like spreadsheets. Unlike Excel spreadsheets, instead of numbering the columns along the side with 1, 2, 3, 4, we use an ID that keeps track of each row of data that's unique.

Instead of putting A, B, C, D, we label the data, employee ID, last name, first name, and so forth.

Let's look at some actual tables using the special tool we have called phpMyAdmin.

NOTE: PHPMyAdmin is downloaded with the WAMP or MAMP software I will show you how to install in Chapter 2.

	ssn	lastname	firstname	phone	address	city	state	zip
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park	CA	94025
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #411	Oakland	CA	94618
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av. #14	San Jose	CA	95128
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	274-80-9391	Straight	Dick	415 834-2919	5420 College Av.	Oakland	CA	94609
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	341-22-1782	Smith	Meander	913 843-0462	10 Misisipi Dr.	Lawrence	KS	66044
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St.	Berkeley	CA	94705
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA	95428
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	486-29-1786	Locksley	Chastity	415 585-4620	18 Broadway Av.	San Francisco	CA	94130

Figure 1.2.1 - PHPMyAdmin showing some author table data

We're looking at the author table. And you can that there's data in there. Ignore these things to the left. This is just to maintain the data with this special tool.

Here you see a social security number, last name, first name, etc., for authors.

Look at the graphic below to see a book table...

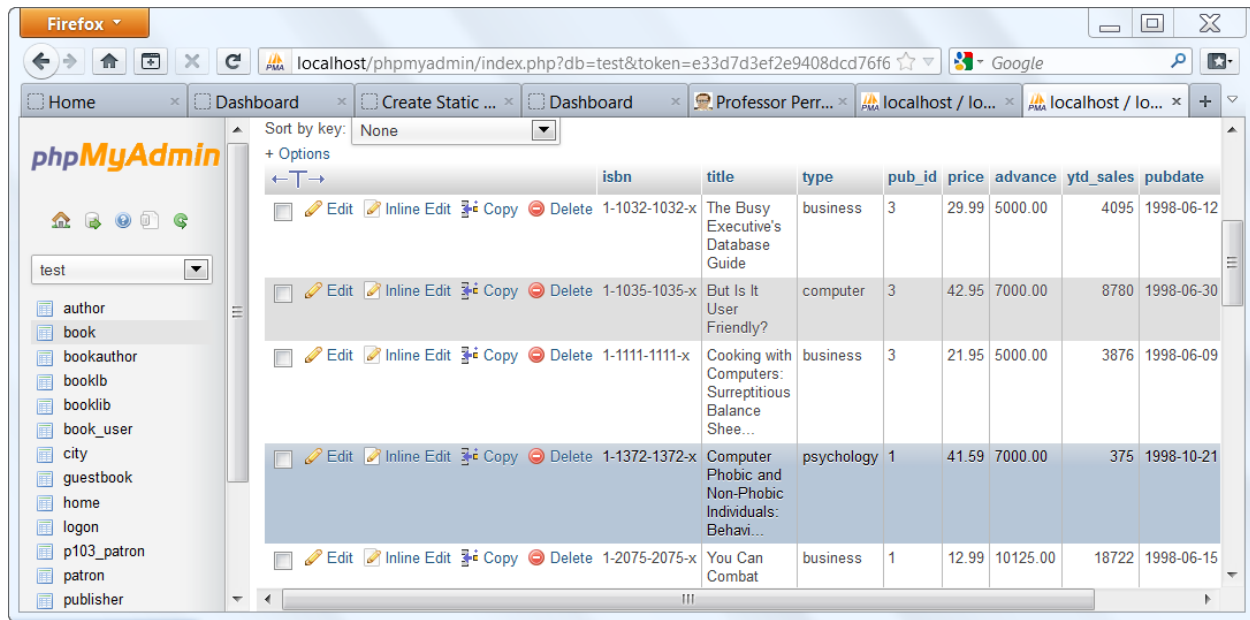


Figure 1.2.2 – PHPMyAdmin showing some book table data

You now see the kind of data that's related to books; ISBN number, titles, what kind of book it is, etc.

While it is nice to be able view the data in these tables interactively, PHP in has special functions that work really well with MySQL and you are going to be able to use PHP to deal with this data in these grid-like formats called tables and use this information to display information on our web pages.

Very powerful stuff and you'll see more as the class progresses.

1.3 Simple PHP Program

This section is on using a simple PHP program. Let's see the page in a browser...

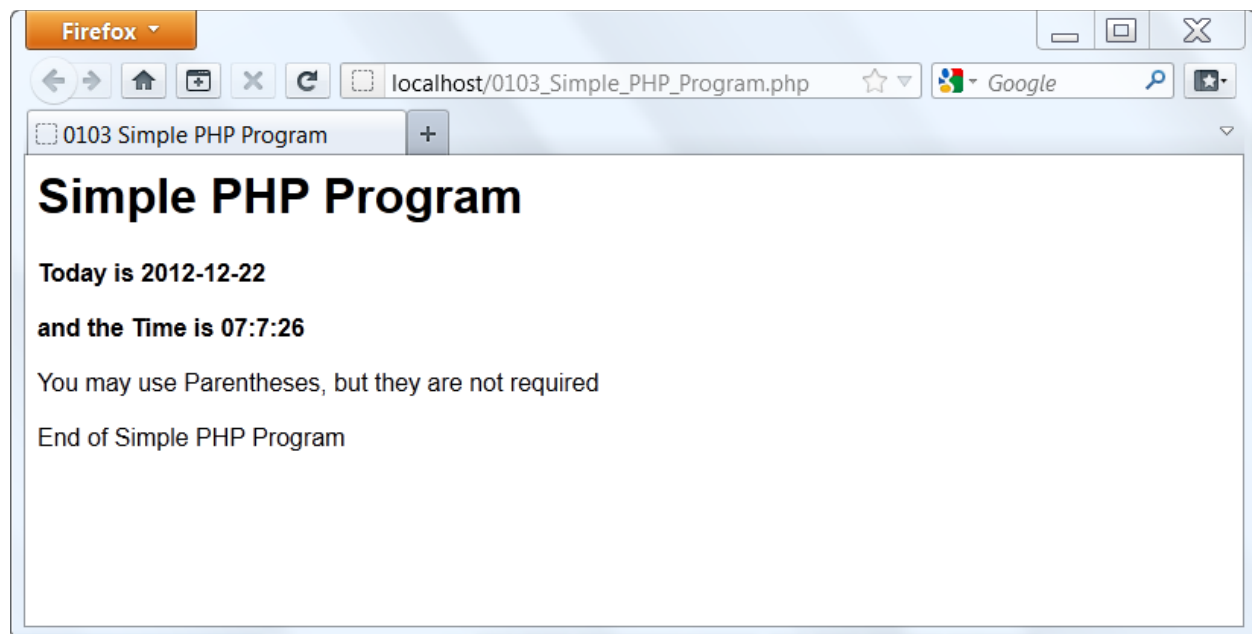


Figure 1.3.1 – Simple PHP Web Page

And you can see from the heading on the page that it is a simple PHP program.

Notice the address bar. You see "localhost/0103_Simple_PHP_Program.php". The "localhost" indicates that you are running this program on a local server (i.e. on your computer) and that the program "0103_Simple_PHP_Program.php" is stored directly under the "root" directory.

NOTE: If you had uploaded the program to a web server that anyone on the Internet could access, the address might be something like this:

http://www.somesite.com/0103_Simple_PHP_Program.php

The page displays a heading "Simple PHP Program, today's date and the time when I ran the web page.

The sentence "You may use parenthesis but they are not required" will be illustrated when I show you the code.

Now, the program has already been run, I want to run that again. When I run it again watch the time as I continue to reload the page.

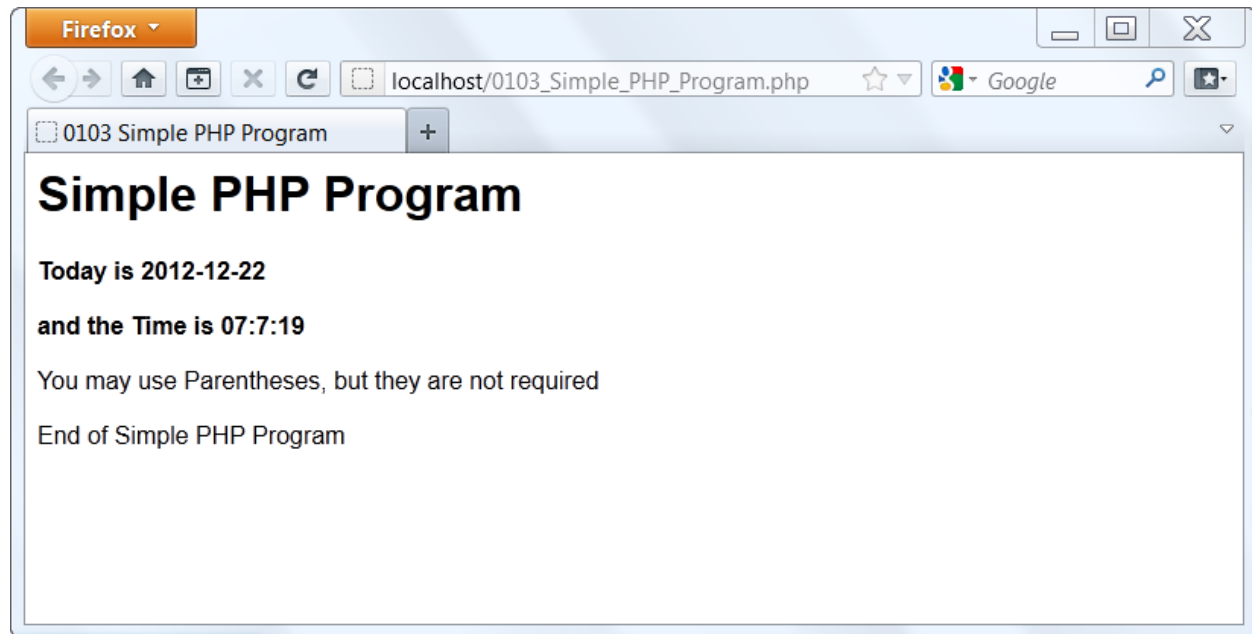


Figure 1.3.2 - Simple PHP Web Page (run a 2nd time)

The seconds keep going up.

So, it's like a little clock but I have to keep reloading the page to get to run. Each time I do this, it runs.

Notice over that the program name itself and has a ".php" extension. This is required for PHP to run, and I'll show you what that will do for you when we look at the code.

Let's look at the code...

```
<html>

<head><title>0103 Simple PHP Program</title>

<style>body {font-family: Arial, Helvetica, sans-serif; font-
size: 16px;}</style>

<body>
```



```
<h1>Simple PHP Program</h1>

<?php

    $today = date('Y-m-d');

    print "<p><b>Today is $today </b></p>";

    $time = date('h:g:s');

    echo "<p><b>and the Time is $time </b></p>";

    print("<p>You may use Parentheses, ");

    echo("but they are not required</p>");

?>

<p>End of Simple PHP Program</p>

</body>

</html>
```

Figure 1.3.3 - Simple PHP Code

The PHP code looks a lot like webpage code... good reason for that... it is the code for a webpage.

A static web page only has HTML code in it. The dynamic (i.e. changing) web page code you will be learning here will have PHP code in it as well.

The HTML code you see here is simplified; I didn't use a **DOCTYPE** or any other tags that are part of the current W3C standard. In real world web pages, you should make your web pages validate to current standards. For this book, I am just to show you just enough code so that you would know to use it is a fully-formed web page.

Back to the program code example ...

All of the HTML code here is just processed normally, nothing changes.

I even applied a *style* here on the body to get different family and font size at 16-point. If you know about styles, you'll be able to do that and I'm going to show a little bit about styles as we proceed through the class. Just a little bit. Just enough to help us format our pages in a little bit nicer way.

Now, where is the PHP code? PHP code is always placed between a the beginning PHP tag (**<?php**) and the ending PHP tag (**?>**). NOTE: These tags and the PHP code found between them is referred to as a *PHP Block* or *Block of Code*

The end result of PHP code is usually just to render HTML code onto the page as if we typed it there to begin with.

You might think, "Well, why don't we just type it there to begin with?"

Because any display that could change in some way needs to be generated with PHP code.

This very first PHP statement runs a **date** function...

```
$today = date('Y-m-d');
```

This is special PHP construct and it says, "Bring back today's year, month, and day", and store it into a PHP variable called `$today`.

PHP always uses a \$ as the first character in a variable name.

Remember what you used to do in grade school where you had..

$x = 5$

$y = 7$

What is $x + y$?

You do the same kind of thing with programming but you use more meaningful variable names.

Instead of...

$x = 2012-12-22$

you would specify...

```
$today = "2012-12-22";
```

Since 'today' describes the kind of data that is stored in the variable and 'x' does not.

So, the next statement here prints the paragraph tag, the bold tag, the words "Today is \$today", and the ending bold tag and ending paragraph tag...

```
print "<p><b>Today is $today </b></p>";
```

NOTE: The variable `$today` will be replaced with the value stored in it when it displays in the browser window. (e.g. "Today is 2012-12-22")

Look at the next line...

```
$time = date('h:g:s');
```

Here's that **date** function again, this time, it's going to bring back hours, minutes and seconds, and it puts in a variable called `$time`.

In the next line I used something different. Instead of the word "**print**" that I used to print (i.e. display on the browser window) I used the word '**echo**'...

```
echo "<p><b>and the Time is $time </b></p>";
```

There is no important difference between a **print** and an **echo** statement. For all practical purposes, they do the same thing. People favor different choices based on what they're used to. Some languages use **echo** more, some use **print**. I favor **print** because I learned Perl first and Perl uses **print**.

Notice in this **echo** statement, I do not put the quoted text in between parentheses.

In the next **print** and **echo** statements, I do...

```
print("<p>You may use Parentheses, ");
```

```
echo("but they are not required</p>");
```

Using parentheses is entirely optional with these statements. It is simply a style choice.

Notice that I starting the paragraph in the **print** statement and then completed the paragraph in the next **echo** statement. That's fine. When I displayed the page, it appeared on one line, because the generated HTML code would make it appear on one line.

Let's go back to the webpage for just a second. You can always do a View-Page-Source to see the generated HTML code. I'm running the page on Firefox, but all major browsers allow this. This is a good way to check if your PHP code is generating the HTML output you think it is.

NOTE: A user of your web pages will never see the PHP code using View-Page-Source. Everything that's rendered back to the browser is only the generated HTML code, never the PHP code. This makes PHP secure.

You secure things on the server that the end user never gets to see. You get the effect of PHP code but never see the PHP code itself. This is one of the important things about all server side languages.

PHP is I probably the easiest server-side language to learn and use.