

PHP – Writing PHP scripts & using Variables R. Berdan

(Last updated Jan 27, 2005)

Prerequisites to complete this module:

- 1) Access to a server running PHP
- 2) Alternatively install Apache server and PHP on Windows
- 3) Need to know how to code some basic HTML
- 4) Need to download and install FTP program and learn how to upload and download files to a server
- 5) Each student should have their own account on a server running PHP

1.1 Introduction

PHP is a server side, HTML embedded, cross platform scripting language. It was first called Personal Home Page in 1994, but was later changed (for marketing reasons) to Hypertext Preprocessor language. It was originally created by Rasmus Lerdorf to track the visitors to his online resume. The language has since been extended considerably, and the current version is 5.0 The syntax of the language is based on C, Java and Perl.

There are several reasons why you might want to learn this scripting language:

- 1) runs faster than CGI, ASP or JSP
- 2) It's free and widely supported (ask your Internet Provider)
- 3) It is easier to learn than Perl and other server side scripting languages

Here is a list of some of the things you can use PHP for:

- 1) Create access counters
- 2) Process forms, send to e-mail, text file or database
- 3) Permit visitors to upload files to your server
- 4) Create calendars
- 5) Create Bulletin Board messaging systems
- 6) Create mailing lists
- 7) Create Shopping carts and online auctions
- 8) Create web site search engines
- 9) Create games
- 10) Post News
- 11) Create instant messaging services
- 12) Pass word protect a site
- 13) Allow users to create and edit web page through the web
- 14) Create Guest books
- 15) Store and retrieve information in the form of cookies
- 16) Conduct surveys or take votes on the web

To use PHP you need to know how to code in HTML, how to FTP your files to a host account, and the host account must support PHP. Many, but not all hosting companies offer PHP because of its many advantages and it is free. Because it is a scripting language, you do not need to compile the script – simply embed the scripts inside your HTML page and/or create standalone php pages and upload them to the sever. Other scripting languages e.g. javascript can also be embedded into HTML – however, the main difference between the two is that javascript is a client side scripting language and PHP is a server side scripting language. Certain things such as processing of forms, counters etc. require a server side component – and PHP can fill this function. PHP was written specifically for making dynamic web sites. Using Active server pages (ASP) requires an understanding of VB scripting and using CGI requires knowledge of Perl. Finally, PHP is not limited to creating only web pages but can be used to make standalone programs, generate graphics and PDF files on the fly.

Finally: PHP is a service that the web server has to provide. Most free web-page hosts, and ISPs giving space will not support PHP. In Calgary Telus does support PHP, but SHAW does not . Shaw’s tech service indicated – security as the main reason. PHP can run anywhere and can be used to create SPAM – multiple e-mails send to large list of people.

Free Hosting with PHP site: <http://www.free-webSPACE.ca> (Jan 2005)

Creating PHP – all you need is Notepad or any simple text editor. You can also use a variety of programmers, text editors that are freely available on the Net.

1) First PHP page; Open notepad and type in the code below.

```
<html>
<head>
<title>My first PHP script</title>
</head>
<body>
<?php

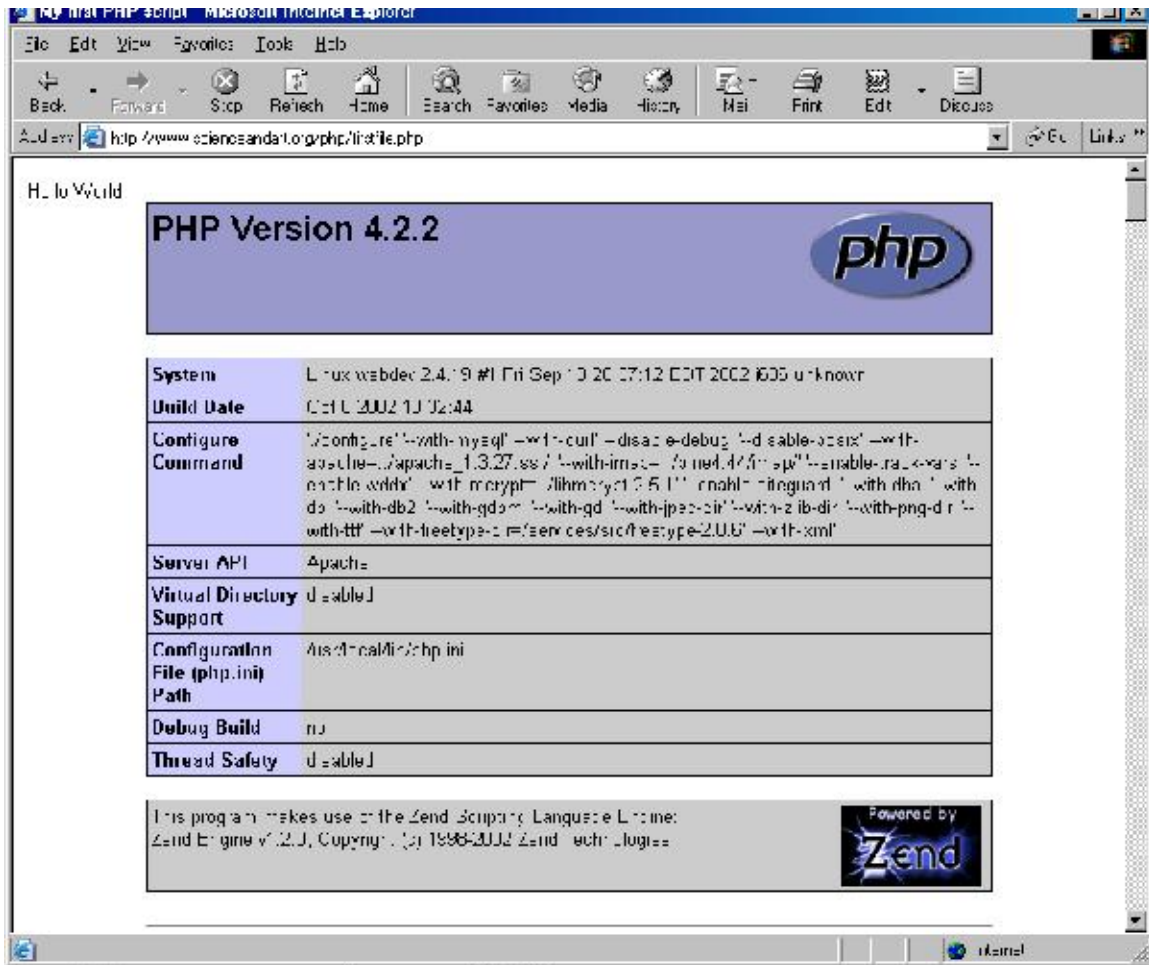
print "Hello World";
phpinfo();

?>

</body>
</html>
```

Save this page as **firstfile.php**, then open up an FTP tool like WS_FTP and upload the file to a PHP enable server. Type in the URL/file.php and the script will display inside your web browser. You may want to put the scripts into a subfolder called php in which case add the file path e.g.

www.scienceandart.org/php/firstfile.php - next page see the screen shot.



At the top left of the browser window is the text “Hello World” and several tables with the PHP version running on your machine.

If you did not see this screen - make sure:

- 1) Your server is in fact PHP enabled (turned On Control panel>Administrative tools>Personal Web Manager (PWM)– Click on the PWM and check if a button in the Main window says stop – then it is on, if not click on the button to turn your PWM on and retest your script.

- 2) make sure added a semicolon to the end of both lines “;” - this is essential
- 3) Check for any other errors you might have made in either your HTML or php script, upload and try again.

If you don't have an FTP program, I recommend you download WS_FTP from <http://www.ftpplanet.com/> - select the LE English version – it is free for students. You will need to know our host URL, 1) ftp address 2) ftp user identification and 3) ftp password for your account or one provided for you.

If you are running server software on your machine you can download and install PHP and run the scripts locally.

1. 2 PHP syntax

To embed PHP inside HTML you need to include the start and end tags

<?php

statements;

?>

Above is the most common way to insert php, there are other methods but the php.ini file must be configured to permit the alternative coding tags e.g.

ASP style

<%

statements;

%>

<SCRIPT LANGUAGE="php">

statements;

</SCRIPT>

I recommend using the first method and will use them exclusively in the course i.e. <?php statements; ?>

In the script we wrote

```
phpinfo();
```

This is a built in function that returns the current version of php running on the server. In PHP functions are not case sensitive and this could be written as PHPINFO(): or Phpinfo();

Sending a TEXT to the browser window

```
<?php
```

```
print "Hello World";
```

```
?>
```

There are several methods for doing this:

```
print "Hello world <br />";  
echo "This is the echo statement <br />";  
print ("this used brackets<br />");
```

the parentheses are not required so I don't normally use them, the
 tag simply adds a carriage return - it is an HTML break tag, it is not required it only ensures that each statement is on a separate line when displayed in the browser.

Printf(format [args]) is frequently used to output currency values with two decimal places. E.g. try the following inside your php tags

```
<?php  
$amount = 24.3956;  
printf ("%01.2f", $amount);  
?>
```

The ""%01.2f" tells PHP to print \$amount using 0 to pad extra spaces, with 1 digit to the left of the decimal and 2 digits to the right of the decimal output = 24.40
Change to printf("%04.3f", "\$amount); output = 0024.396.

To add the \$ sign to the amount use printf("\$%04.2f", \$amount); output = \$0024.40

Embedding HTML inside print statements

```
<?php
```

```
print "<center><h1>I am Canadian</h1></center>";
```

```
?>
```

```
<?php
```

```
print "<font size=\\'7\\'>I am Canadian</font>";
```

```
?>
```

HTML tags that require quotation marks must be escaped using the backslash ("\"), the is will print the quotation instead of interpreting it.

SEMICOLONS are mandatory at the end of statements!

One of the most common mistakes by beginning programmers is to leave the semicolons out at the end of the statements. In javascript, semicolons are optional but not in PHP!

While semicolons must always be added to the end of PHP statements NEVER put a semicolon at the end of a conditional test e.g. if (age > 16); or a loop e.g. for (i=0; i>4; i++); - if you do it will result in an error.

WHITESPACE

White space is generally (but not universally) ignored and any blank line or spaces, or tab is ignored.

NEW LINE Command \n

```
<?php
```

```
print "I am Canadian! \n");
```

```
print "What Nationality are you?";
```

```
?>
```

\n new line command, \t tab – visible when viewing data in wordpad

```
<html>
<head>
<title>Adding a backslash</title>
</head>
<body>
<?php

print "I am Canadian! \n";
print "What Nationality are you?";

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

save as addingbackslash.php - type in URL and run the script. What did you see? You should have seen all the text in one line without a break! \n only adds space in a text file or e-mail not HTML. If you want to add space in HTML you will need to add a break tage
 at the end of the statement inside the quotes. Note the script works fine without the additional HTML around the code.

We will return to this command later when we send text to an email address or a text file on the server

Adding a **TAB \t** will add tabs to your php scripts to create spacing in text files (not HTML pages). \r carriage return; \t tab

COMMENTS

Putting in comments is essential in many programs even to the original programmer who may forget what their train of thought was when they made the program and have to return to the script several weeks or months later to modify or update the script. **DO NOT put semicolons at the end of your comments.**

There are 3 ways to add comments to PHP

```
<?php

// single line comment
/* multiline comment
is done the same way
as it is in javascript */
# comment follows a hash like in Perl
```

Try it – add several comments, upload and preview the page.

```
<html>
<head>
<title>Adding a backslash</title>
</head>
<body>

<?php

print "I am Canadian! <br>\n";
print "What Nationality are you?";

// single line comment
# this is another way to leave a single line comment
/* multiline comment on 2 or more lines
is done the same way in javascript
*/

?>

</body>
</html>
```

Save upload file and run script then select View Source code in the browser - you should see the following code below. Note the php tags are not visible, the comments are not visible – they are only readable in the original text file.

```
<html>
<head>
<title>Adding a backslash</title>
</head>
<body>

I am Canadian! <br>
What Nationality are you?
</body>
</html>
```

CASE - variable names are case sensitive in PHP, but not the functions.

```
<?php
```

```
PHPINFO():
```

```
phpinfo();
```

```
PhpInfo();
```

```
?>
```

All of these will yield the same result

1.3 HOW PHP WORKS

Viewer requests a PHP page or link ---- the Server → PHP → HTML → Client

A call to the server to the file.php , the PHP file is parsed and creates an HTML page dynamically and sends it back to the client . For this reason the php script is not visible,nor are the comments since the script is parsed on the server then sent to the browser. Javascript in contrast executes or parses within the browser on the client's computer – not the server.

See Text L. Ullman PHP for the World Wide Web, page XIV for a diagram that shows how PHP works.

1.4 PHP Variables (See Chapter 2 in Text book)

Variables are names we provide to access data stored in the computers memory.

Rules for naming variables in PHP

1. All variable names are preceded with a \$sign e.g. \$variablename
2. Variable names are case sensitive
3. Variables names must include A-Z, a-z or the underscore character
4. Variables names can not contain spaces or special characters e.g. \, ?, * etc
5. You do not need to declare a variable before using it (loosely typed)
6. You do not have to initialize the variable before using it
7. No size limit on the number of characters a variable name can have

Examples of valid variable names:

\$color
\$operating_system
\$_modelnumber

Examples of invalid variable names:

\$this&that	can not add symbol &
\$!encounter	can not add symbol !
\$4infinity	can not have a number precede

Types of Variables in PHP

1. Numbers integers (1, -2), floating point (2.03), scientific notation – but not fractions (1\4) e.g. scientific notation 0.314E2 = 31.4
2. Strings – text e.g. \$name = "Fred"; or name='Fred'; single quote different result – PHP will output \$name - i.e. the variable name not the value.
3. Booleans – true, false (true=1, false=0)
4. Objects – PHP Objects are data types that allow for storage of the data and also the information on how to process it. Data elements = properties, how to process data = methods
5. Predefined variables (environment variables) – built in.
6. Arrays – multiple values associated with a single variable name

```

<html>
<head>
<title>Adding a backslash</title>
</head>
<body>

<?php
$FirstName = "Fred";
print "Hello $FirstName<br>";
// prints to the browser Hello Fred

print 'Hello $FirstName <br>';
// prints Hello $FirstName;

print "Hello \$FirstName <br>";
// prints Hello $FirstName escaped the variablename

?>
</body>
</html>

```

Note the difference between using single & Double Quotes !

```

<?php

$actor = "Marlon Brando";
print "$actor";
// prints Marlon Brando

echo $actor;
// also prints Marlon Brando

print $actor;
// also prints Marlon Brando

print '$actor';
// prints $actor

?>

```

Everything stored \$variablename = "4" becomes a string even numbers

Type Casting

Casting involves forcing a variable to behave as a specific variable type. This is accomplished by placing the intended type in front of the variable to be cast. A type can be cast by inserting (type) in front of the variables.

Cast Operators

(int) or (integer)
(real) (double) or (float)
(string)
(array)
(object)

Conversion

Integer
Double (float)
String
Array
Object

Example

```
<?php
```

```
$variable1 = "4.0";  
$variable2= 5;  
$variable3 = (int)$variable1 + (string)$variable2;  
print "$variable3"; // outputs 9
```

```
?>
```

```
<?php
```

```
$variable1 = "4.0";  
$variable2= 5;  
$variable3 = (string)$variable1 + (string)$variable2;  
print "$variable3"; // still outputs 9 because of the + symbol, try . symbol  
// you should see a value of 4.05 with dot concatenation symbol
```

```
?>
```

```
<?php
```

```
$variable1 = 4.0;  
$variable2= 5;  
$variable3 = $variable1 * $variable2;  
print "$variable3"; // outputs 20 even if you type cast values as strings
```

```
?>
```

```
<?php
$variable1 = 15.6;
$variable2 = (int)$variable1;
print "$variable2";
// outputs 15 converts number to integer

?>
```

```
<?php
$variable1 = 5;
$variable2="100 bottles of beer on the wall";
$variable3 = $variable1 + $variable2;
print "$variable3"; // output 105!

?>
```

The reason is that the PHP parser determines the type by looking at only the initial part of a string. If you were to change `$variable2 = " There are 100 bottles of beer on the wall";` it will output a value of 5 because the string would evaluate to 0.

```
<?php
$variable1 = 3;
$variable2= 5.4;
$variable3 = $variable1 + $variable2;
print "$variable3"; // converts to float or double and outputs 8.4!

?>
```

This behavior of the variables is called **"Type Juggling"**.

Variable Variables

On occasion is useful to make use of variables whose contents can be treated dynamically as a variable itself.

```
$recipe = "spaghetti;
$$recipe = "& meatballs"; /* this assigns "& meatballs" to a variable named
                             spagehetti*/
print $recipe . " " . $$recipe;
// prints spagehetti & meatballs
```

Predefined Variables (also called Environment Variables).

Predefined variables provide the developer with information about the server configuration. PHP creates some of the variables while others are created depending on the operating system and web server PHP is running. The variables are always typed in CAPS. You can view them using `phpinfo()`;

Environment

Variable	Value
ALLUSERSPROFILE	C:\Documents and Settings\All Users
CommonProgramFiles	C:\Program Files\Common Files
COMPUTERNAME	SMOKING
ComSpec	C:\WINNT\system32\cmd.exe
CONTENT_LENGTH	0
GATEWAY_INTERFACE	CGI/1.1
HTTPS	off
HTTP_ACCEPT	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, application/x-shockwave-flash, *
HTTP_ACCEPT_LANGUAGE	en-us
HTTP_CONNECTION	Keep-Alive
HTTP_HOST	localhost
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
HTTP_ACCEPT_ENCODING	gzip, deflate
INSTANCE_ID	1
LOCAL_ADDR	127.0.0.1
NUMBER_OF_PROCESSORS	1
Os2LibPath	C:\WINNT\system32\os2dll;
OS	Windows_NT
Path	C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem;C:\Program Files\Common Files\Sonic Shared\Ligos\GoMotion;C:\Program Files\Common Files\Sonic Shared\Ligos\Decoders;C:\Program Files\Common Files\Sonic Shared>MainConcept
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PATH_INFO	/test.php
PATH_TRANSLATED	c:\inetpub\wwwroot\test.php
PROCESSOR_ARCHITECTURE	x86
PROCESSOR_IDENTIFIER	x86 Family 15 Model 2 Stepping 7, GenuineIntel
PROCESSOR_LEVEL	15
PROCESSOR_REVISION	0207
ProgramFiles	C:\Program Files
REMOTE_ADDR	127.0.0.1
REMOTE_HOST	127.0.0.1
REQUEST_METHOD	GET
SCRIPT_NAME	/test.php
SERVER_NAME	localhost
SERVER_PORT	80

Determine your server IP address:

```
<?php
print "Hi you IP address is: $REMOTE_ADDR";
?>
```

Try printing to the screen don't forget to add \$

\$OS - to determine your operating system

\$HTTP_USER_AGENT - to determine the browser type

\$COMPUTER_NAME – to determine the name of the computer on a network

Constants

A constant is a value that cannot be modified through out the execution of the program. In PHP, constants are defined use the define() function. Once defined it can not be changed (or redefined) at any other point of the program.

E.g. define ("Pi", "3.141592")

```
<?php
```

```
define("Pi", "3.14592");
print "The value of Pi is " . Pi ; // don't forget . concat symbol before Pi
```

```
?>
```

Another Example of defining a constant – a constant is referred to without quotes.

```
<?php
define ("Freezing_Point", "0 celsius");
print "Freezing_Point";
// this will print the variable name Freezing_Point
?>
```

```
<?php
define ("Freezing_Point", "0 celsius");
print Freezing_Point;
// This will print 0 celsius!
?>
```

PHP has several functions to determine the current data type of a variable – specifically **gettype()** and **settype()**

```
<?php  
  
$number = " 5.0";  
echo gettype($number)  
// string  
  
?>
```

```
<?php  
  
$number = 5.0;  
echo gettype($number)  
// double  
  
?>
```

```
<?php  
  
$number = 5;  
echo gettype($number)  
// integer  
  
?>
```

Isset, unset, empty Functions

isset() determines whether a variable has been created – it is used frequently to determine whether a person has submitted a form e.g.

```
isset($submit) ; // checks to see if submit button has been clicked on.  
unset($number); // destroys a variable and releases the memory allocated to it  
empty($number); // returns 1 or true if there is no variable or the variable =0;
```

In Summary

Two things in using variables that are quite different in PHP than javascript:

- 1) Variables enclosed within double quotes are still treated as variables not strings
- 2) Variables surround by single quotes or have an escape character before them “ \ “ will be displayed as the \$variable name not the variable value.

Exercise working with PHP variables :

```
<html>
<head>
<title>Convert</title>
</head>
<body>

<?php
$Enginetype = "2.0L";
$Taxrate = 3;
$Taxpaid = $Enginetype * $Taxrate;
print "Engine Type: $Enginetype<br>";
print "Tax Rate: $Taxrate<br>";
print "Tax Paid: $Taxpaid";

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

What will the output be?

Answer:

```
Engine Type: 2.0L
Tax Rate: 3
Tax Paid: 6
```

Explanation

PHP does not care that there is an L in the variable \$Enginetype – it simply grabs the value of 2.0 at the beginning of the data and uses it for the calculation.