



tutorials

Cookies
ASP
Advanced HTML
Frames & Tables
.htaccess
Rounded Table Corners
Flash 5
FTP
Beginners
Javascript
PHP/MySQL
Site Promotion
Mobile Internet: WML/WAP
Server Side Includes (SSI)
HTML - The Basics
XHTML
Stylesheets
XML
PHP

related links

Hotscripts
Official PHP Home Page
PHP Hosts at Free-Webhosting.info
More PHP Sites

PHP Tutorial

Part 2 - Displaying Information & Variables

- Part 1 - Introduction
- **Part 2 - Displaying Information & Variables**
- Part 3 - IF Statements
- Part 4 - Loops and Arrays
- Part 5 - E-mail With PHP
- Part 6 - PHP With Forms
- Part 7 - Final Notes

Introduction

In the last part of the tutorial I explained some of the advantages of PHP as a scripting language and showed you how to test your server for PHP. In this part I will show you the basics of displaying information in the browser and how you can use variables to hold information.

Printing Text

To output text in your PHP script is actually very simple. As with most other things in PHP you do it in a variety of different ways. The main one you will be using, though, is print. Print is used to output text, variables or a combination of the two so that they display on the screen.

The print statement is used in the following way:

```
print("Hello world!");
```

I will explain the above line:

print is the command and tells the script what to do. This is followed by the information to print, which is contained in the brackets. Because you are outputting text, the text is also enclosed in quotation marks. Finally, as with nearly every line in a PHP script, it must end in a semicolon. You would, of course, have to enclose this in your standard PHP tags, making the following code:

```
<?
print("Hello world!");
?>
```

Which will display:

Hello world!

on the screen.

Variables

As with other programming languages, PHP allows you to define variables. In PHP there are many variable types, but the most common is called a String. It can hold text and numbers. All variables begin with a \$ sign. To assign some text to a string you would use the following code:

MEDWAY® Va
Manufacturer of '
W100,13x6 - 55>

This is quite a simple line to understand, everything inside the quotation marks will be assigned to the string. You must remember a few rules about strings though:

Strings are case sensitive so \$Welcome_Text is not the same as \$welcome_text
String names can contain letters, numbers and underscores but cannot begin with a number or underscore
When assigning numbers to strings you do not need to include the quotes so:

`$user_id = 987`

would be allowed.

Outputting Variables

To display a variable on the screen uses exactly the same code as to display text but in a slightly different form. The following code would display your welcome text:

```
<?
$welcome_text = "Hello and welcome to my website.";
print($welcome_text);
?>
```

As you can see, the only major difference is that you do not need the quotation marks if you are printing a variable.

Formatting Your Text

Unfortunately, the output from your PHP programs is quite boring. Everything is just output in the browser's default font. It is very easy, though, to format your text using HTML. This is because, as PHP is a server side language, the code is executed before the page is sent to the browser. This means that only the resulting information from the script is sent, so in the example above the browser would just be sent the text:

Hello and welcome to my website.

This means, though, that you can include standard HTML markup in your scripts and str only problem with this is that many HTML tags require the " sign. You may notice that th with the quotation marks used to print your text. This means that you must tell the script quotes should be used (the ones at the beginning and end of the output) and which ones ignored (the ones in the HTML code).

For this example I will change the text to the Arial font in red. The normal code for this w

```
<font face="Arial" color="#FF0000">
</font>
```

As you can see this code contains 4 quotation marks so would confuse the script. Because you must add a backslash before each quotation mark to make the PHP script ignore it. would change to:

```
<font face=\"Arial\" color="#FF0000\">
</font>
```

You can now include this in your print statement:

```
print("<font face=\"Arial\" color=\"#FF0000\">Hello and welcome to my website.</font>");
```

which will make the browser display:

Hello and welcome to my website.

because it has only been sent the code:

```
<font face="Arial" color="#FF0000">Hello and welcome to my website.</font>
```

This does make it quite difficult to output HTML code into the browser but later in this tut show you another way of doing this which can make it a bit easier.

Part 3

In part 3 I will introduce If statements.

- [Part 1 - Introduction](#)
- [Part 2 - Displaying Information & Variables](#)
- [Part 3 - IF Statements](#)
- [Part 4 - Loops and Arrays](#)
- [Part 5 - E-mail With PHP](#)
- [Part 6 - PHP With Forms](#)
- [Part 7 - Final Notes](#)
- [Hotscripts](#)
- [Official PHP Home Page](#)
- [PHP Hosts at Free-Webhosting](#)
- [More PHP Sites](#)
- [Related Reading](#)

© 1999 - 2001 David Gowans

