PHP Tutorial

Part 4 - Loops and Arrays

- 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

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

Introduction

In the last parts of this tutorial I have showed you how to deal with text and variables in P how you can use IF statements to compare them and to make decisions. In this part I ar show you how to use another important part of PHP, loops.

The WHILE Loop

The WHILE loop is one of the most useful commands in PHP. It is also quite easy to set A WHILE loop will, as the name suggests, execute a piece of code until a certain conditi

Repeating A Set Number Of Times

If you have a piece of code which you want to repeat several times without retyping it, yo while loop. For instance if you wanted to print out the words "Hello World" 5 times you co following code:

```
$times = 5;
$x = 0;
while ($x < $times) {
echo "Hello World";
++$x;
}
```

I will now explain this code. The first two lines are just setting the variables. The $times v holds the number of times you want to repeat the code. The $x variable is the one which the number of times the code has been executed. After these is the WHILE line. This tel computer to repeat the code while $i is less than $times (or to repeat it until $i is equal to This is followed by the code to be executed which is enclosed in { }.

After the echo line which prints out the text, there is another very important line:

```
++$x;
```

What this does is exactly the same as writing:

```
$x = $x + 1;
```

It adds one to the value of $x. This code is then repeated (as $x now equals 1). It continues being repeated until $x equals 5 (the value of times) when the computer will then move on to the next part of the code.

Using $x

The variable counting the number of repeats ($x in the above example) can be used for much more than just counting. For example if you wanted to create a web page with all the numbers from 1 to 1000 on it, you could either type out every single one or you could use the following code:

```
$number = 1000;
$current = 0;
while ($current < $number) {
++$current;
echo "$current<br>";
}
```

There are a few things to notice about this code. Firstly, you will notice that I have placed the ++$current; before the echo statement. This is because, if I didn't do this it would start printing numbers from 0, which is not what we want. The ++$current; line can be placed anywhere in your WHILE loop, it does not matter. It can, of course, add, subtract, multiply, divide or do anthing else to the number as well.

The other reason for this is that, if the ++$current; line was after the echo line, the loop would also stop when the number showed 999 because it would check $current which would equal 1000 (set in the last loop) and would stop, even though 1000 had not yet been printed.

Arrays

Arrays are common to many programing languages. They are special variables which can hold more than one value, each stored in its own numbered 'space' in the array. Arrays are extremely useful, especially when using WHILE loops.

Setting Up An Array

Setting up an array is slightly different to setting up a normal variable. In this example I v
array with 5 names in it:

```
$names[0] = 'John';
$names[1] = 'Paul';
$names[2] = 'Steven';
$names[3] = 'George';
$names[4] = 'David';
```

As you can see, the parts of an array are all numbered, starting from 0. To add a value t
you must specify the location in the array by putting a number in [ ].

Reading From An Array

Reading from an array is just the same as putting information in. All you have to do is to
array and the number of the piece of data in the array. So if I wanted to print out the thir
could use the code:

n

```
echo "The third name is $names[2]";
```

Which would output:

The third name is Steven

Using Arrays And Loops

One of the best uses of a loop is to output the information in an array. For instance if I w
print out the following list of names:

Name 1 is John
Name 2 is Paul
Name 3 is Steven
Name 4 is George
Name 5 is David

I could use the following code:

```
$number = 5;
$x = 0;
while ($x < $number) {
$namenumber = $x + 1;
echo "Name $namenumber is $names[$x]<br>";
++$x;
}
```

As you can see, I can use the variable $x from my loop to print out the names in the arra
have noticed I am also using the variable $namenumber which is always 1 greater than
because the array numbering starts from 0, so to number the names correctly in the outp
add one to the actual value.

Part 5

In the next part I will show you how you can send e-mail using PHP.

- 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-Webhostin
- More PHP Sites
- Related Reading