PHP Tutorial

Part 6 - PHP With Forms

Introduction

In the last part, I showed you how to use PHP to send e-mail messages using a script. I
will continue this and also show you how to use PHP and forms together to make your P
useful.

Setting Up Your Form

Setting up a form for use with a PHP script is exactly the same as normal in HTML. As tl
tutorial I will not go into depth in how to write your form but I will show you three of the m
of code you must know:

```
<input type="text" name="thebox" value="Your Name">
```

Will display a text input box with Your Name written in it as default. The value section of
optional. The information defined by name will be the name of this text box and should b

```
<textarea name="message">
Please write your message here.
</textarea>
```

Will display a large scrolling text box with the text 'Please write your message here.' as c
Again, the name is defined and should be unique.

```
<input type="submit" value="Submit">
```

This will create a submit button for your form. You can change what it says on the buttor
changing the button's value.

All the elements for your form must be enclosed in the <form> tags. They are used as fo

```
<form action="process.php" method="post">
Form elements and formatting etc.
</form>
```

The form's action tells it what script to send its data to (in this case its process.php). This
be a full URL (e.g. http://www.mysite.com/scripts/private/processors/process.php). The r
the form how to submit its data. POST will send the data in a data stream to the script w
requested. GET is the other option. GET will send the form data in the form of the url so

appear after a question mark e.g. http://www.mysite.com/process.php?name=david

It really makes no difference which system you use but it is normally better to use POST if you are using passwords or sensitive information as they should not be shown in the browser's address bar.

Getting The Form Information

The next step is to get the data the form has submitted into your script so that you can do something with it. This is. There are basically two different methods of getting the data into PHP, which depend on how they were submitted. There are two submission methods, GET and POST, which can both be used by forms. The difference between the two is that using GET, the variables and data will be shown in the page address, but using POST it is invisible. The benefit of GET, though is that you can submit information to the script without a form, by simply editing the URL.

This works the same as submitting a form using GET. The advantage of this is that you can create links to your scripts which do different things depending on the link clicked. For example you could create a script which will show different pages depending on the link clicked:

```
yourpage.php?user=david
```
could show David's page and:
```
yourpage.php?user=tom
```
could show Tom's page, using the same script.

It is also possible to pass more than one piece of information to the script using this system by separating them with the & symbol:

```
yourpage.php?user=david&referrer=gowansnet&area=6
```

These could all be accessed separately using the GET variables user, referrer and area.

To get a variable which has been sent to a script using the POST method you use the following code:
```
$variablename=$_POST['variable'];
```
which basically takes the variable from the POST (the name of a form field) and assigns it to the variable $variablename.

Similarly, if you are using the GET method you should use the form:
```
$variablename=$_GET['variable'];
```

This should be done for each variable you wish to use from your form (or URL).

Creating The Form To Mail Script

To finish off this section, I will show you how to use what you have learnt in this part and create a system which will e-mail a user's comments to you.

Firstly, create this form for your HTML page:

```
<form action="mail.php" method="post">
Your Name: <input type="text" name="name"><br>
E-mail: <input type="text" name = "email"><br><br>
Comments<br>
<textarea name="comments"></textarea><br><br>
<input type="submit" value="Submit">
</form>
```

This will make a simple form where the user can enter their e-mail address, their name a
comments. You can, of course, add extra parts to this form but remember to update the
Now create the PHP script:

```
<?
function checkOK($field)
{
if (eregi("\r",$field) || eregi("\n",$field)){
die("Invalid Input!");
}
}

$name=$_POST['name'];
checkOK($name);
$email=$_POST['email'];
checkOK($email);
$comments=$_POST['comments'];
checkOK($comments);
$to="php@gowansnet.com";
$message="$name just filled in your comments form. They said:\n$comments\n\nTheir e
address was: $email";
if(mail($to,"Comments From Your Site",$message,"From: $email\n")) {
echo "Thanks for your comments.";
} else {
echo "There was a problem sending the mail. Please check that you filled in the form co
}
?>
```

Remember to replace php@gowansnet.com with your own e-mail address. This script sl
saved as mail.php and both should be uploaded. Now, all you need to do is to fill in your
form.

The first part of that script may look a bit strange:

```
function checkOK($field)
{
if (eregi("\r",$field) || eregi("\n",$field)){
die("Invalid Input!");
}
}
```

You don't really need to worry about what this is doing, but basically, it stops spammers
your form to send thier spam messages by checking special characters are not present i
which can be used to trick the computer into sending messages to other addresses. It is
which checks for these characters, and if they are found, stops running the script.

The lines:

```
checkOK($name);
```

etc. run this check on each input to ensure it is valid.

Part 7

As you can see, using forms with PHP can be very effective. In the next part I will show y
the final things you should know about PHP.

- Part 1 - Introduction
- Part 2 - Displaying Information & Variables
- Part 3 - IF Statements
- Part 4 - Loops and Arrays
- Part 5 - E-mail With PHP

- Hotscripts
- Official PHP Home Page
- PHP Hosts at Free-Webhostin
- More PHP Sites
- Related Reading