form follows function

or function follows forms? ;)

goals for the day

You should have created a form that we can add some functionality to...

and you have already completed the Form Validation tutorial as well.

There are two methods to retrieve that information from your users...

1) via secure email utilizing PHP

2) utilizing php along with a mySQL database

server-side processing

server-side processing is both
interactive and dynamic

CGI (Computer Gateway Interface)

protocol (method) for a web server to pass a request
to an application, process the data and send back
a message to the user.

server-side scripting

server-side scripting is a technology where the server-side script is embedded within a web page document, such as a .php or .asp file.

it is different from CGI in that it uses direct execution, meaning the script is run by the server.

each time you perform a search with a search engine, you are utilizing server-side processing.

steps in server-side processing

(1) Web page invokes server-side processing by a form action attribute or by a hyperlink

(2) web server executes server-side script

(3) server-side script accesses requested database, file or process

(4) web server returns web page with requested info or confirmation of action

server-side scripting

PHP

(hypertext preprocessing)

server-side scripting language

PHP scripts are executed on the server

can contain text, html tags and scripts

Why PHP?

runs on different platforms (windows, linux, unix)

open source & free to download

compatible with almost all servers

Why PHP?

designed to be used alongside html...
php and html are interchangeable within the page

can break apart html document
in order to streamline revisions...
change one file, it changes everywhere

PHP syntax

A PHP scripting block always starts with
<?php and ends with ?>

Anything outside the php tags is read as html

A PHP scripting block can be placed
anywhere in the document.

server-side scripting

when utilizing a server-side script,
the code must communicate about the
form method attribute (get or post), and the
form action attribute (url of the server-side script).

The value of the name attribute on each form control
is passed to the server-side script and may be used
as a variable name in the server-side processing.

get method passes the info in the url (not secure)
post method passes info in the body of the http request

server-side scripting

let's look at an example:

```
4   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5   <title>sgk feedback email form</title>
6   </head>
7   <body>
8
9   <!--this is the message the user will see once the form is submitted. you can make
    interesting AND it should reflect the look of your site as well as keeping navigat
    consistent with the rest of the site-->
10  <h1>Thank you for your response. </h1>
11
12  <!--begin php-->
13  <?php
14
15  //insert your own email address between the quotation marks
16  $mailto = "melissaflicker@gmail.com";
17
18  //insert the subject of the email between the quotation marks
19  $subject = "Feedback form";
20
21  //this formats the beginning of the message, change what is between the quotes if
22  $message = "Values submitted from web site form:";
23
24  //if you have a textbox with name/id of "email" it will use that email address to
    it will come from your site. the below piece of code states that the header of the
    take its value from the textbox called "email" on your html form
25  $header = "From: ".$_POST['email'];
26
27
28  //this is the code that takes each value from your html form and places it inside
    the email.
29  foreach ($_POST as $key => $value)
30  {
31     if (!is_array($value))
32     {
33        $message .= "\n".$key." : ".$value;
34     }
35     else
36     {
37        foreach ($_POST[$key] as $itemvalue)
38        {
39           $message .= "\n".$key." : ".$itemvalue;
40        }
41     }
42  }
43
44  //the following code is what sends the email and compiles all of the above info
45  mail($mailto, $subject, stripslashes($message), $header);
46
47
48  //end php
49  ?>
```

remember our goals?

Today we are going to use the following method to retrieve that information from your users...

1) via secure email utilizing PHP

using php to collect form data via email

now let's try it...

we are going to configure a php document that will send you an email with the information from your form.

how is this different from using
a mailto: action in your html form?

Two big differences:
- user can submit without using an email client
- more secure than mailto: action

using php to collect form data via email

(1) create a new document called feedback.php and save it in the same folder as your form

(2) find the html form that you created for today. look at the opening <form> tag. make sure the action is set to "feedback.php" and the method is set to "post"

... add a text field called "email" (if you don't already have one)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>sgk email form test</title>
</head>

<body>
<form action="feedback.php" method="post" name="feedbackform">

<label>name</label>
<input name="name" type="text" size="20" /> <br />


<label>email</label>
<input name="email" type="text" maxlength="50" />
<p>
  <label>
    <input type="checkbox" name="favorite_dessert" value="ice cream" id="favorite_dessert" />
    ice cream</label>
  <br />
  <label>
    <input type="checkbox" name="favorite_dessert" value="cake" id="favorite_dessert" />
    cake</label>
  <br />
</p>

<input name="" type="submit" />

</form>


</body>
</html>

```

using php to collect form data via email

(3) go back to your document called feedback.php

let's look at the code that will add the functionality

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>sgk feedback email form</title>
</head>
<body>

<!--this is the message the user will see once the form is submitted. you can make this much more
interesting AND it should reflect the look of your site as well as keeping navigation, header, etc.
consistent with the rest of the site-->
<h1>Thank you for your response. </h1>

<!--begin php-->
<?php

//insert your own email address between the quotation marks
$mailto = "yourname@domain.com";

//insert the subject of the email between the quotation marks
$subject = "Feedback form";

//this formats the beginning of the message, change what is between the quotes if you like
$message = "Values submitted from web site form:";

//if you have a textbox with name/id of "email" it will use that email address to send the form, otherwise
 it will come from your site. the below piece of code states that the header of the email you receive will
 take its value from the textbox called "email" on your html form
$header = "From: ".$_POST['email'];


//this is the code that takes each value from your html form and places it inside the message or body of
the email.
foreach ($_POST as $key => $value)
{
   if (!is_array($value))
   {
      $message .= "\n".$key." : ".$value;
   }
   else
   {
      foreach ($_POST[$key] as $itemvalue)
      {
         $message .= "\n".$key." : ".$itemvalue;
      }
   }
}

//the following code is what sends the email and compiles all of the above info
mail($mailto, $subject, stripslashes($message), $header);


//end php
?>

</body>
</html>
```

using php to collect form data via email

(4) modify feedback.php accordingly

save both the .html form and the .php feedback page and upload.

visit your form and test. give it a few minutes and you should receive an email with the results.

notice how the info that you put into the .html email form is returned to you. also notice the subject line, message and how it corresponds to the code in the.php file.

so that's awesome....

Now what?

Your challenge:

Design a critique form for your site that will allow others to assess your website design and functionality as it evolves.

The form is useless unless it works, so you will have to code it so that it will email you appropriate responses.... hurray.

2) utilizing php along with a mySQL database