Introduction to PHP

What is PHP?

It's official name is PHP: Hypertext Preprocessor, and it is a server-side scripting language. When your Web browser accesses a URL, it is making a request to a web server. When you request a PHP page, for example, http://fog.ccsf.edu/~srubin/formA.php, the Web server wakes up the PHP parsing engine and says, "Hey! You've got to do something before I send a result back to this person's Web browser."  
  
Then the PHP parsing engine runs through the PHP code found in formA.php, and returns the resulting output. This output is passed back to the Web server as part of the HTML code in the document, which in turn is passed to your browser, which displays it to you.

What does PHP do?

Here are some common uses of PHP:

* Perform system functions: create, open, read from, write to, and close files on your system; execute system commands; create directories; and modify permissions.
* Gather data from forms: save the data to a file, send data via e-mail, return manipulated data to the user.
* Access databases and generate content on-the-fly, or create a web interface for adding, deleting, and modifying elements within your database.
* Set cookies and access cookie variables.
* Start sessions and use session variables and objects.
* Use PHP user authentication to restrict access to sections of your web page.
* Create images on-the-fly.
* Encrypt Data.

The Form statement and PHP directory:

In Homework 8b, you are to create a form and a php script for processing the form, namely, (1) to send the form information from the user back to you in an email, and (2) to return the form information back to the user's screen as a verification that the email has been sent.  
  
The form is a normal HTML page with form elements. The form statement should contain an action equal to your php filename along with a method of post. For example:  
  
<form action="formA.php" method="post">  
  
For the Hills server: Your php file must reside in a directory, called php, which is a sub-directory of your public\_html directory. The php directory should have a unix permission of 711, that is, read permission for owner and execute permission for owner, other, and group. A permission of 755 will also work. Be sure to upload your php file in ascii transfer mode.  
  
Regarding the form statement, notice that the action's value does not refer to your php directory, because the server will know to automatically look in the php directory for any files having a file extension of .php. You must NOT include the reference to the php directory in the path; the special cgiwrap that Apache uses when it sees the .php extension presumes that directory implicitly. Note that other servers may not require your php files to be in a special sub-directory called php.  
  
Please Note: your php directory should be directly within your public\_html directory.  
  
If your hw8b.html has as its URL:   
  
http://hills.ccsf.edu/~USERID/cnit132/hw8b.html  
  
then the php directory is 1 directory up from hw8b.html and your form statement would be:  
  
<form action="../formA.php" method="post">  
  
If your hw8b.html has as its URL:   
  
http://hills.ccsf.edu/~USERID/cnit132/homework/hw8/hw8b.html  
  
then the php directory is 3 directories up from hw8b.html and your form statement would be:  
  
<form action="../../../formA.php" method="post">

The PHP script to email your form and return a screen verification:

Your PHP script must have a file extension of .php. Here is the content of my formA.php.

<?php $msg = "E-MAIL SENT FROM FOG\n";

$msg .= "Sender's Name:

$\_POST[sender\_name]\n";

$msg .= "Sender's E-mail:

$\_POST[sender\_email]\n";

$msg .= "Sender's Age:

$\_POST[age]\n"; $msg .= "Message:

$\_POST[message]\n\n";

$to = "srubin45@comcast.net";

$subject = "Form Feedback from Fog";

$mailheaders = "From:

$\_POST[sender\_email]\n";

$mailheaders .= "Reply-To:

$\_POST[sender\_email]\n\n";

mail($to, $subject, $msg, $mailheaders); ?>

<html>

<head>

<title>The following email has been sent to Steve Rubin</title>

</head>

<body bgcolor="#f5f5dc">

<p><b>Your Name: </b>

<?php echo

"$\_POST[sender\_name]"; ?> <br /><br />

<b>Your E-mail Address: </b>

<?php echo "$\_POST[sender\_email]"; ?> <br /><br />

<b>Your Age: </b> <?php echo

"$\_POST[age]"; ?> <br /><br />

<b>Your Message:</b>

<?php echo "$\_POST[message]"; ?>

</p>

</body>

</html>

Explanation of the above code:

The code in formA.php will send an email back to you and and return the form information back to the user's screen.  
  
A. Begin a PHP block and start building a message string:

<?php  
$msg = "E-MAIL SENT FROM FOG\n";   
  
NOTE: The message E-MAIL SENT FROM FOG will be the first line in the message text of the email that is sent and can be changed to anything you like. \n is a carriage return.   
  
  
B. Continue building the message string by adding an entry for the sender's name:

$msg .= "Sender's Name: $\_POST[sender\_name]\n";

NOTE: The .= indicates concatenation. Your variable after $\_POST must match exactly the variable name in your form input text box for sender's name. For example in my HTML form page, I have:  
  
<input type="text" name="sender\_name" size="40">   
  
NOTE: You don't have to use the variable sender\_name in your form or in this php file. What matters is that the names must MATCH.   
  
C. Continue building the message string by adding an entry for the sender's e-mail address, an entry for age, and an entry for the message:   
  
$msg .= "Sender's E-mail:   
$\_POST[sender\_email]\n";   
$msg .= "Sender's Age:  
$\_POST[age]\n";   
$msg .= "Message:   
$\_POST[message]\n\n";   
  
NOTE: The extra spaces before $\_POST are there so that the messages will be better aligned vertically when you are reading the email. The variable names after $\_POST (in this case, sender\_email, age, and message) must match exactly the variable name in your form input text box, radio buttons, and textarea. For Example, in my HTML form page, I have:  
  
<input type="text" size="25" name="sender\_email" />   
<input type="radio" name="age" value="0-20" />Under 21   
<textarea rows="5" cols="20" name="message"></textarea>   
  
D. Create a variable to hold the recipient's e-mail address (substitute your own):  
  
$to = "srubin45@comcast.net";   
  
E. Create a variable to hold the subject of the e-mail:  
  
$subject = "Form Feedback from Fog";   
  
NOTE: You can change the string inside the quotes to anything you like.  
  
  
F. Create a variable to hold additional mailheaders:   
  
$mailheaders = "From: $\_POST[sender\_email]\n";   
  
NOTE: This will put the sender's email address in the from field of the sent email that comes to you. Note also that the variable name inside of the square brackets must match exactly the variable name in your form input text box for sender's email address.  
  
  
G. Add to the $mailheaders variable:  
  
$mailheaders .= "Reply-To: $\_POST[sender\_email]\n\n";   
  
NOTE: the variable name inside of the square brackets must match exactly the variable name in your form input text box for sender's email address.  
  
  
H. Add the mail() function:  
  
mail($to, $subject, $msg, $mailheaders);   
  
I. Close your PHP block:

?>

NOTE: You're not done yet. Although this code will send the email, you should return something to the user's screen so that they know the form has been sent. Otherwise, they might sit there and continually click the Send button.  
  
J. The next 6 lines are normal HTML statements.  
  
K. Add the text label for the Your Name field and display the user's input:  
  
<p><b>Your Name: </b>   
<?php echo "$\_POST[sender\_name]"; ?>   
  
NOTE: sender\_name must match exactly the variable name in your form input text box for sender's name.   
  
<?php echo "$\_POST[sender\_name]"; ?>   
returns to the screen the value of that form name variable.   
  
  
L. Add the text label for the Your E-mail Address field and display the user's input:  
  
<b>Your E-mail Address: </b>   
  
<?php echo "$\_POST[sender\_email]"; ?>   
  
NOTE: sender\_email must match exactly the variable name in your form input text box for sender's email.  
  
  
M. Add the text label for the Age field and display the user's input:  
  
<b>Your Age: </b>   
<?php echo "$\_POST[age]"; ?>   
  
NOTE: age must match exactly the variable name in your form input text box for sender's age.  
  
  
N. Add the text label for the Message field and display the user's input:  
  
<b>Your Message:</b>   
<?php echo "$\_POST[message]"; ?>  
  
NOTE: message must match exactly the variable name in your form textarea for message.  
  
  
O. Add the closing HTML statements.

PHP Tutorials:

<http://us2.php.net/tut.php>   
  
<http://www.w3schools.com/php/default.asp>  
  
<http://www.freewebmasterhelp.com/tutorials/php>