



# PHP Tutorial: Writing Your First PHP Script: Feedback Form Script

by Christopher Heng, [thesitewizard.com](http://thesitewizard.com)

I have always believed that the most fun way to learn a new programming language, whether it is a language like C or a scripting language like PHP, is to use it to write a real-life useful program. Of course this is not the most systematic method of learning, but it works well if you already have some background in programming.

## Preliminaries

1. Before you write your first PHP program, make sure that that your website is running on *a web host* that runs PHP 4.1 or above.
2. You may also find it useful to have a copy of PHP 4.1 or later installed on your own computer. This makes testing your PHP scripts much easier. If you use Windows, you can find some tips on installing PHP on your own computer from my article on "[How to Install and Configure PHP 5 to Run with Apache on Windows](http://www.thesitewizard.com/php/install-php-5-apache-windows.shtml)" at <http://www.thesitewizard.com/php/install-php-5-apache-windows.shtml>
3. And of course, you will need *an ASCII text editor* of some kind (such as Notepad on Windows). There's a list of such editors on <http://www.thefreecountry.com/programming/editors.shtml>
4. This tutorial also assumes that you have at least some knowledge of HTML. This is necessary because if I have to explain all the HTML tags as well, this tutorial will wind up being tediously long.

I will begin with a very rudimentary (but working) PHP script to take input from a feedback form and send it to you in an email message. This type of form is sometimes referred to as a FormMail or Form to Mail script. In later articles, I will develop that script (and others) to include features commonly found in such FormMail scripts.

If you are programming-savvy, you will recognize this as a sort of "Hello World" program, but infinitely more useful!

## Writing the Feedback Form

The first thing we need to do is to write the feedback form itself. Put the following code in the <body> section of an HTML file named, say, feedback.html.

```
<form method="post" action="sendmail.php">
  Email: <input name="email" type="text" /><br />
  Message:<br />
  <textarea name="message" rows="15" cols="40">
</textarea><br />
```

```
<input type="submit" />
</form>
```

Basically the form asks the visitor for his email address (the field named "email" found in `input name="email"` above) and message (the field named "message" found in `textarea name="message"`), and presents him with a button which he can click to submit the contents of the form. When the form is submitted, it is "posted" (see the "method" attribute of the `<form>` tag) to a script named "sendmail.php" (also specified in the `<form>` tag).

## The Feedback Form PHP Script

Now all that remains is to code "sendmail.php". This is made extremely easy by the facilities available in PHP. Type the following code into a file named "sendmail.php". Do not put anything else into that file, ie, don't put in any other HTML tags or headers, etc.

```
<?php
    $email = $_REQUEST['email'] ;
    $message = $_REQUEST['message'] ;

    mail( "yourname@example.com", "Feedback Form Results",
        $message, "From: $email" );
    header( "Location: http://www.example.com/thankyou.html" );
?>
```

When the form is submitted to sendmail.php, the contents of the "email" field in the form is put into a PHP variable called `$_REQUEST['email']`. Likewise the contents of the "message" field is put into the variable `$_REQUEST['message']`.

If you had named the fields in your form "emailofsender" and "contentsofmessage", then the information submitted in those fields would have been available to your script in the variables `$_REQUEST['emailofsender']` and `$_REQUEST['contentsofmessage']` respectively. I think you get the idea.

The first thing we do in our PHP script is to make the information that is submitted easily accessible to the rest of the program.

Firstly, we made a copy of the contents of `$_REQUEST['email']` in a variable we call `$email`. This was done in the line

```
$email = $_REQUEST['email'] ;
```

Note that we don't really have to call this new variable `$email` - we could have called it `$thingamajig` if we wished, but it makes sense to name a variable with some meaningful name.

Likewise, in the next line, we made a copy (assigned) of `$_REQUEST['message']` in a variable `$message`.

```
$message = $_REQUEST['message'] ;
```

Again, we could have named the new variable anything we wanted - but it's easier for us to understand the program if the variable name reflects what it does.

The real workhorse of this script is in the line beginning with "mail".

```
mail( "yourname@example.com", "Feedback Form Results",  
      $message, "From: $email" );
```

`mail` is a special function in PHP that sends mail. The first parameter to `mail` is supposed to contain the email address you want the form contents to be sent to, such as your own email address. The second parameter is the "Subject" of the email message. The last two parameters are the content of the message and the headers you want sent, respectively. We want a "From" header so that we know who is sending the email to us and can reply to him/her if we need to.

Notice that, like many other programming languages, strings (sequences of characters) are enclosed in double quotes, such as "Feedback Form Results".

Variables like `$message` can be used as-is. Note also that you can also interpolate (introduce) the contents of the variable `$email` into a string, like "From: `$email`", so that if your `$email` string contained an address like "william@shakespeare.com", the final string that is passed to the `mail` function will be "From: william@shakespeare.com".

You can also use single quotes (such as those in 'Hi there') to quote strings, but when you do so, the variables included are not expanded. This is useful if, for some reason, you really want to pass the string 'From: `$email`' to `mail` without PHP translating that to "From: william@shakespeare.com".

You can't interpolate variables like `$_REQUEST['email']` and `$_REQUEST['message']` in the same way, which is why we assigned their contents to `$email` and `$message`. The latter are also easier to read anyway.

Finally, it is appropriate to thank the visitor for his message. This is done with the line

```
header( "Location: http://www.example.com/thankyou.html" );
```

This line causes PHP to send an HTTP header back to the visitor's browser telling it to load the URL "http://www.example.com/thankyou.html". The "header" function allows us to send any HTTP header to the browser.

You will of course have to create such a file called "thankyou.html" with some sort of message to thank your visitor for his efforts, otherwise your visitor will be greeted with an unfriendly "404/File Not Found" error after he sends his message. You should also replace the URLs and email addresses with the correct ones if you want to use that script on your site.

By the way, the script has to be enclosed within the "<?php" and ">" tags because the PHP processor treats all input as HTML code unless otherwise specified. On most systems, you can simply use "<?" and ">" as the opening and closing tags to get the script to work,

however if you want to be sure that your script will work on all systems, you should use the full "<?php" form for the opening tag.

Easy wasn't it? In just a few lines, you've written your first PHP script. And it's not some trivial and useless script - it is actually a working, usable program!

Before you put the script "live" on the Internet, check out the following additional tutorials:

- In *[How to Improve Your Form to Mail Script](#)*, you will develop the script so that your visitor's input is checked to catch instances where someone accidentally clicks the "Submit" button before they fill in their email address. You will also learn how to integrate both the form and the script into a single "feedback.php" file if you wish.
- The article *[How to Prevent Email Injection in Your PHP Form to Mail Scripts](#)* deals with the security aspects of putting a PHP form to mail script "live" on the Internet, where spammers can abuse it and hijack it to send spam to others.

Copyright 2000-2007 by Christopher Heng. All rights reserved.

Get more free tips and articles like *[this](#)*, on web design, promotion, revenue and scripting, from *<http://www.thesitewizard.com/>*

If you find this article useful, please consider making a donation.

[Make A Donation](#)

You are here: *[Top](#)* > *[Free PHP Tutorials: PHP Programming, PHP Installation and Configuration](#)* > *[PHP Tutorial: Writing Your First PHP Script: Feedback Form Script](#)*

Other articles on: *[PHP](#)*

## thesitewizard™ News Feed (RSS Site Feed)

Do you find this article useful? You can learn of new articles and scripts that are published on *[thesitewizard.com](http://www.thesitewizard.com)* by subscribing to the RSS feed. Simply point your RSS feed reader or a browser that supports RSS feeds at *<http://www.thesitewizard.com/thesitewizard.xml>*. You can *[read more about how to subscribe to RSS site feeds](#)* from my *[RSS FAQ](#)*.

## Please Do Not Reprint This Article

This article is copyrighted. Please do not reproduce this article in whole or part, in any form, without *[obtaining my written permission](#)*.

## Related Pages

- *[How to Install and Configure PHP 5 to Run with Apache on Windows](#)*

- [How to Install and Configure PHP4 on Windows](#)
- [How to Protect Your Images from Bandwidth Thieves \(PHP Script\)](#)
- [How to Install and Configure Apache 2 on Windows](#)
- [Introduction to Cascading Style Sheets \(CSS\)](#)
- [How to Accept Credit Card Payments from Your Website](#)
- [Which Web Host Would You Recommend? \(FAQ\)](#)
- [Frame Breakout JavaScript for your website \(Wizard\)](#)
- [thefreecountry.com's Free PHP Scripts](#)

## New Pages

- [How to Create a Website with Dreamweaver CS4 \(Dreamweaver Tutorial\)](#)
- [How to Install Apache 2.2 on Windows Vista](#)
- [How to Uninstall the KompoZer Web Editor from Your Windows Computer](#)
- [How to Create a Rectangular Box to Contain Your Text/Pictures with CSS](#)
- [Why Don't You Use the Keywords Meta Tag on Your Website?](#)
- [How to Add a Background Image to Your Web Page in KompoZer and Nvu](#)
- [Review of Amaya: W3C's Free WYSIWYG Web Authoring Environment](#)
- [What is HTML, CSS, JavaScript, PHP and Perl? Do I Need to Learn Them to Create a Website?](#)
- [How Much Does It Cost to Set Up a Website?](#)
- [How to Design a Two Column Layout for Your Website Using CSS](#)

## Popular Pages

- [How to Start / Create Your Own Website: The Beginner's A-Z Guide](#)
- [Tips on Choosing a Good Domain Name](#)
- [How to Create a Search Engine Friendly Website](#)
- [How to Create a Website with Dreamweaver CS4 \(Dreamweaver Tutorial\)](#)
- [How to Design and Publish Your Website with KompoZer \(free WYSIWYG web editor\)](#)

## How to Link to This Page

To link to this page from your website, simply cut and paste the following code to your web page.

```
<a  
href="http://www.thesitewizard.com/archive/  
feedbackphp.shtml" target="_top">PHP  
Tutorial: Writing Your First PHP Script:  
Feedback Form Script</a>
```

It will appear on your page as:

[PHP Tutorial: Writing Your First PHP Script: Feedback Form Script](#)

Copyright © 2000-2008 by Christopher Heng. All rights reserved.  
thesitewizard™, thefreecountry™ and HowToHaven™ are trademarks of Christopher Heng.  
Last updated: 5 May 2008.

