



Solar Energy Charity

Helping Relieve Poverty Through The Provision of Solar Energy. J

Public Service Ads

[tutorials](#)
[articles](#)
[tools](#)
[templates](#)
[directory](#)
[ask](#)
[newsletter](#)


[Sitemap](#) | [Contact](#) | [Link To Us](#) | [Advertise](#)
[Report A Problem](#)
[Home : Tutorials](#)

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](#)

PHP Tutorial

Part 5 - E-mail With 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](#)

Introduction

One of the major uses of a server side scripting language is to provide a way of sending the server and, in particular, to take form input and output it to an e-mail address. In this show you how to send e-mail messages using PHP.

The Mail Command

Mail is extremely easy to send from PHP, unlike using scripting languages which require setup (like CGI). There is actually just one command, `mail()` for sending mail. It is used as

```
mail($to,$subject,$body,$headers);
```

In this example I have used variables as they have descriptive names but you could also use text in the mail command. Firstly, `$to`. This variable (or section of the command) contains the address to which the mail will be sent. `$subject` is the section for the subject of the e-mail. `$body` is the actual text of the e-mail.

The section `$headers` is used for any additional e-mail headers you may want to add. The common use of this is for the From field of an e-mail but you can also include other headers like `Cc` and `bcc`.

Sending An E-mail

Before sending your mail, if you are using variables, you must, of course, set up the variables beforehand. Here is some simple code for sending a message:

```
$to = "php@gowansnet.com";
$subject = "PHP Is Great";
$body = "PHP is one of the best scripting languages around";
$headers = "From: webmaster@gowansnet.com\n";
mail($to,$subject,$body,$headers);
echo "Mail sent to $to";
```

This code will actually do two things. Firstly it will send a message to `php@gowansnet.com` with the subject 'PHP Is Great' and the text:

PHP is one of the best scripting languages around

and the e-mail will be from webmaster@gowansnet.com. It will also output the text:

Mail sent to php@gowansnet.com

to the browser.

Formatting E-mail

Something you may have noticed from the example is that the From line ended with \n. This is actually a very important character when sending e-mail. It is the new line character and tells PHP to take a new line in an e-mail. It is very important that this is put in after each header you add so that your e-mail will follow the international standards and will be delivered.

The \n code can also be used in the body section of the e-mail to put line breaks in but should not be used in the subject or the To field.

Mail Without Variables

The e-mail above could have been sent using different variable names (it is the position of the variables in relation to the commas, not the name of them which decides on their use). It could also have been done on one line using text like this:

```
mail("php@gowansnet.com","PHP Is Great","PHP is one of the best
scripting languages around","From: webmaster@gowansnet.com\n");
```

But that would make your code slightly harder to read.

Error Control

As anyone who has been scripting for a while will know, it is extremely easy to make mistakes in your code and it is also very easy to input an invalid e-mail address (especially if you are using your script for form to mail). Because of this, you can add in a small piece of code which will check if the e-mail is sent:

```
if(mail($to,$subject,$body,$headers)) {
echo "An e-mail was sent to $to with the subject: $subject";
} else {
echo "There was a problem sending the mail. Check your code and make sure that the e-mail
address $to is valid";
}
```

This code is quite self explanatory. If the mail is sent successfully it will output a message to the browser telling the user, if not, it will display an error message with some suggestions for the problem.

Part 6

In part 6 I will continue covering mail by showing you how to make a simple form to mail with 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-Webhosting.com](#)
- [More PHP Sites](#)
- [Related Reading](#)

- [Part 6 - PHP With Forms](#)
- [Part 7 - Final Notes](#)

© 1999 - 2001 David Gowans

NO SETUP FEE!!!