

24. PHP Sending E-mails

PHP allows you to send e-mails directly from a script.

24.1. The PHP mail() Function

The PHP mail() function is used to send emails from inside a script.

Syntax

```
mail(to, subject, message, headers, parameters)
```

Parameter	Description
to	Required. Specifies the receiver / receivers of the email
subject	Required. Specifies the subject of the email. Note: This parameter cannot contain any newline characters
message	Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters
headers	Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n)
parameters	Optional. Specifies an additional parameter to the sendmail program

Note: For the mail functions to be available, PHP requires an installed and working email system. The program to be used is defined by the configuration settings in the php.ini file. Read more in our PHP Mail reference.

24.2. PHP Simple E-Mail

The simplest way to send an email with PHP is to send a text email.

In the example below we first declare the variables (\$to, \$subject, \$message, \$from, \$headers), then we use the variables in the mail() function to send an e-mail:

```
<?php
$to = "someone@example.com";
$subject = "Test mail";
$message = "Hello! This is a simple email message.";
$from = "someoneelse@example.com";
$headers = "From:" . $from;
mail($to,$subject,$message,$headers);
```

```
echo "Mail Sent.";  
?>
```

24.3. PHP Mail Form

With PHP, you can create a feedback-form on your website. The example below sends a text message to a specified e-mail address:

```
<html>  
<body>  
  
<?php  
if (isset($_REQUEST['email']))  
//if "email" is filled out, send email  
{  
    //send email  
    $email = $_REQUEST['email'] ;  
    $subject = $_REQUEST['subject'] ;  
    $message = $_REQUEST['message'] ;  
    mail("someone@example.com", $subject,  
        $message, "From:" . $email);  
    echo "Thank you for using our mail form";  
}  
else  
//if "email" is not filled out, display the form  
{  
    echo "<form method='post' action='mailform.php'>  
    Email: <input name='email' type='text'><br>  
    Subject: <input name='subject' type='text'><br>  
    Message:<br>  
    <textarea name='message' rows='15' cols='40'>  
    </textarea><br>  
    <input type='submit'>  
    </form>";  
}  
?>  
  
</body>  
</html>
```

This is how the example above works:

- First, check if the email input field is filled out

- If it is not set (like when the page is first visited); output the HTML form
- If it is set (after the form is filled out); send the email from the form
- When submit is pressed after the form is filled out, the page reloads, sees that the email input is set, and sends the email

Note: This is the simplest way to send e-mail, but it is not secure. In the next chapter of this tutorial you can read more about vulnerabilities in e-mail scripts, and how to validate user input to make it more secure.

24.4. PHP Mail Reference

For more information about the PHP mail() function, visit our [PHP Mail Reference](#).