

PHP Send Emails

In this tutorial you will learn how to send simple text or HTML emails directly from the script using the PHP `mail()` function.

The PHP mail() Function

Sending email messages are very common for a web application, for example, sending welcome email when a user create an account on your website, sending newsletters to your registered users, or getting user feedback or comment through website's contact form, and so on.

You can use the PHP built-in `mail()` function for creating and sending email messages to one or more recipients dynamically from your PHP application either in a plain-text form or formatted HTML. The basic syntax of this function can be given with:

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

The following table summarizes the parameters of this function.

Parameter	Description
Required — The following parameters are required	
<code>to</code>	The recipient's email address.
<code>subject</code>	Subject of the email to be sent. This parameter i.e. the subject line cannot contain any newline character (<code>\n</code>).
<code>message</code>	Defines the message to be sent. Each line should be separated with a line feed-LF (<code>\n</code>). Lines should not exceed 70 characters.
Optional — The following parameters are optional	
<code>headers</code>	This is typically used to add extra headers such as "From", "Cc", "Bcc". The additional headers should be separated with a carriage return plus a line feed-CRLF (<code>\r\n</code>).
<code>parameters</code>	Used to pass additional parameters.

Sending Plain Text Emails

The simplest way to send an email with PHP is to send a text email. In the example below we first declare the variables — recipient's email address, subject line and message body — then we pass these variables to the `mail()` function to send the email.

Example

[Download](#)

```
<?php
$to = 'maryjane@email.com';
$subject = 'Marriage Proposal';
$message = 'Hi Jane, will you marry me?';
$from = 'peterparker@email.com';

// Sending email
if(mail($to, $subject, $message)){
    echo 'Your mail has been sent successfully.';
} else{
    echo 'Unable to send email. Please try again.';
}
?>
```

Sending HTML Formatted Emails

When you send a text message using PHP, all the content will be treated as simple text. We're going to improve that output, and make the email into a HTML-formatted email.

To send an HTML email, the process will be the same. However, this time we need to provide additional headers as well as an HTML formatted message.

Example

[Download](#)

```
<?php
$to = 'maryjane@email.com';
$subject = 'Marriage Proposal';
$from = 'peterparker@email.com';

// To send HTML mail, the Content-type header must be set
$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";

// Create email headers
$headers .= 'From: ' . $from . "\r\n".
    'Reply-To: ' . $from . "\r\n" .
    'X-Mailer: PHP/' . phpversion();

// Compose a simple HTML email message
$message = '<html><body>';
```

```
$message .= '<h1 style="color:#f40;">Hi Jane!</h1>';  
$message .= '<p style="color:#080;font-size:18px;">Will you marry me?</p>';  
$message .= '</body></html>';  
  
// Sending email  
if(mail($to, $subject, $message, $headers)){  
    echo 'Your mail has been sent successfully.';  
} else{  
    echo 'Unable to send email. Please try again.';  
}  
?>
```

Note: However, the PHP `mail()` function is a part of the PHP core but you need to set up a mail server on your machine to make it really work.

In the next two chapters ([PHP Form Handling](#) and [PHP Form Validation](#)) you will learn how to implement an interactive contact form on your website to receive the user's comment and feedback through emails using this PHP send mail feature.