PHP - Validate E-mail

The easiest and safest way to check whether an email address is well-formed is to use PHP's filter\_var() function.

In the code below, if the e-mail address is not well-formed, then store an error message:

$email = test\_input($\_POST["email"]);  
if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {  
  $emailErr = "Invalid email format";  
}

## PHP - Validate Name, E-mail, and URL

Now, the script looks like this:

### Example

<!DOCTYPE HTML>    
<html>  
<head>  
<style>  
.error {color: #FF0000;}  
</style>  
</head>  
<body>    
  
<?php  
// define variables and set to empty values  
$nameErr = $emailErr = $genderErr = $websiteErr = "";  
$name = $email = $gender = $comment = $website = "";  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
  if (empty($\_POST["name"])) {  
    $nameErr = "Name is required";  
  } else {  
    $name = test\_input($\_POST["name"]);  
    // check if name only contains letters and whitespace  
    if (!preg\_match("/^[a-zA-Z-' ]\*$/",$name)) {  
      $nameErr = "Only letters and white space allowed";  
    }  
  }  
    
  if (empty($\_POST["email"])) {  
    $emailErr = "Email is required";  
  } else {  
    $email = test\_input($\_POST["email"]);  
    // check if e-mail address is well-formed  
    if (!preg\_match("/^([a-z0-9\+\_\-]+)(\.[a-z0-9\+\_\-]+)\*@([a-z0-9\-]+\.)+[a-z]{2,6}$/ix",$email)) {  
      $emailErr = "Invalid email format";  
    }  
  }  
      
  if (empty($\_POST["website"])) {  
    $website = "";  
  } else {  
    $website = test\_input($\_POST["website"]);  
    // check if URL address syntax is valid  
    if (!preg\_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~\_|!:,.;]\*[-a-z0-9+&@#\/%=~\_|]/i",$website)) {  
      $websiteErr = "Invalid URL";  
    }      
  }  
  
  if (empty($\_POST["comment"])) {  
    $comment = "";  
  } else {  
    $comment = test\_input($\_POST["comment"]);  
  }  
  
  if (empty($\_POST["gender"])) {  
    $genderErr = "Gender is required";  
  } else {  
    $gender = test\_input($\_POST["gender"]);  
  }  
}

function test\_input($data) {  
  $data = trim($data);  
  $data = stripslashes($data);  
  $data = htmlspecialchars($data);  
  return $data;  
}  
?>  
  
<h2>PHP Form Validation Example</h2>  
<p><span class="error">\* required field</span></p>  
<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">    
  Name: <input type="text" name="name">  
  <span class="error">\* <?php echo $nameErr;?></span>  
  <br><br>  
  E-mail: <input type="text" name="email">  
  <span class="error">\* <?php echo $emailErr;?></span>  
  <br><br>  
  Website: <input type="text" name="website">  
  <span class="error"><?php echo $websiteErr;?></span>  
  <br><br>  
  Comment: <textarea name="comment" rows="5" cols="40"></textarea>  
  <br><br>  
  Gender:  
  <input type="radio" name="gender" value="female">Female  
  <input type="radio" name="gender" value="male">Male  
  <input type="radio" name="gender" value="other">Other  
  <span class="error">\* <?php echo $genderErr;?></span>  
  <br><br>  
  <input type="submit" name="submit" value="Submit">    
</form>  
  
<?php  
echo "<h2>Your Input:</h2>";  
echo $name;  
echo "<br>";  
echo $email;  
echo "<br>";  
echo $website;  
echo "<br>";  
echo $comment;  
echo "<br>";  
echo $gender;  
?>  
  
</body>  
</html>

### Example

Send a simple email:

<?php  
// the message  
$msg = "First line of text\nSecond line of text";  
  
// use wordwrap() if lines are longer than 70 characters  
$msg = wordwrap($msg,70);  
  
// send email  
mail("someone@example.com","My subject",$msg);  
?>

Definition and Usage

The mail() function allows you to send emails directly from a script.

Syntax

mail(*to,subject,message,headers,parameters*);

Parameter Values

Send an HTML email:

<?php  
$to = "somebody@example.com, somebodyelse@example.com";  
$subject = "HTML email";  
  
$message = "  
<html>  
<head>  
<title>HTML email</title>  
</head>  
<body>  
<p>This email contains HTML Tags!</p>  
<table>  
<tr>  
<th>Firstname</th>  
<th>Lastname</th>  
</tr>  
<tr>  
<td>John</td>  
<td>Doe</td>  
</tr>  
</table>  
</body>  
</html>  
";  
  
// Always set content-type when sending HTML email  
$headers = "MIME-Version: 1.0" . "\r\n";  
$headers .= "Content-type:text/html;charset=UTF-8" . "\r\n";  
  
// More headers  
$headers .= 'From: <webmaster@example.com>' . "\r\n";  
$headers .= 'Cc: myboss@example.com' . "\r\n";  
  
mail($to,$subject,$message,$headers);  
?>