

Working with Forms

PHP introduces the most common ways in which user can use PHP to interact with the user, web forms. In total, we'll show you how to use PHP and web forms to carry out the following tasks:

- Pass data from a form to a PHP script
- Validate form data
- Work with multi-valued form components

Introduction to how PHP is able to accept and process data submitted through a web form.



PHP and Web Forms

One of the most powerful features of PHP is the way it handles HTML forms. There are two regular methods for sending data from one script to another script, GET and POST. Although GET is the default, POST method will is capable of handling more data than GET. An important characteristic is, when you're using forms to insert and modify large blocks of text. If you use POST, any posted data sent to a PHP script must be referenced using the `$_POST`.

Example

```
<html>
<body>
<form method= "get" action= "<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type= "text" name= "firstname">
  <input type= "submit" value= "submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "GET") {
  $fname = $_GET['firstname'];
  if (empty($fname)) {
    echo "Name is empty";
  } else {
    echo $fname;
  }
}
```

```
}  
}  
?>  
</body>  
</html>
```

Example 2

```
<html>  
<body>  
<form action="read.php" method="get">  
Full Name: <input type="text" name="name"><br>  
Mobile Number: <input type="text" name="number"><br>  
<input type="submit">  
</form>  
</body>  
</html>
```

read.php

```
<html>  
<body>  
Hi <?php echo $_GET["name"]; ?><br>  
Your mobile number: <?php echo $_GET["number"]; ?>  
</body>  
</html>
```

Output:

Hi Sakshi Education
Your mobile number 9642000000.

Get vs Post

- Both GET and POST create an array and the arrays carry key and value pairs, here keys are the names of the form inputs and values are data passed by the user.
- Both GET and POST are super global variables, so we can use them anywhere of our program, in any class.
- In GET method, it uses URL to pass the values and keys.

- Where as in POST method it uses HTTP Post method to send the data.

PHP Form Validation

The HTML form we will be working with various input fields some of them are required and some are optional text fields, radio buttons and a submit button.

Example

```
<!DOCTYPE HTML>
<html>
<head>
</head>
<body>
<?php
// define and initialize variables and set them to empty
$name = "";
$email = "";
$gender = "";
$message = "";
$age = "";
$namemsg = "";
$emailmsg = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST["name"];
    $email = $_POST["email"];
    $age = $_POST["age"];
    $message = $_POST["message"];
    $gender = $_POST["gender"];

    if($name == ""){
        $namemsg = "required";
    }
    if($email == ""){
        $emailmsg = "required";
    }
}

?>
<h2>PHP Form Validation Example</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

```
Your Name: <input type="text" name="name"> <span style="color:red;">* <?php
echo $namemsg; ?></span><br/><br/>
Your E-mail: <input type="text" name="email"> <span style="color:red;">* <?php
echo $emailmsg; ?></span><br/><br/>
Your Age: <input type="text" name="age"><br/><br/>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male<br/><br/>
Message : <textarea name="message" rows="5" cols="40"></textarea><br/><br/>
<input type="submit" name="submit" value="Submit">
</form>

<?php
if($name != " " || $email != "") {

    echo "<h2>Details you entered:</h2>";
    echo $name;
    echo "<br>";
    echo $email;
    echo "<br>";
    echo $age;
    echo "<br>";
    echo $gender;
    echo "<br>";
    echo $message;
}
?>
</body>
</html>
```

The screenshot shows a web browser window with the address bar displaying 'localhost/sakshipost/index.php'. The page title is 'PHP Form Validation Example'. The form contains the following elements:

- 'Your Name:' followed by a text input field with a red asterisk (*) to its right.
- 'Your E-mail:' followed by a text input field with a red asterisk (*) to its right.
- 'Your Age:' followed by a text input field.
- 'Gender:' followed by two radio buttons labeled 'Female' and 'Male'.
- A large text area for 'Message:'.
- A 'Submit' button at the bottom left.

In above program Name and Email id are required if you click on submit without entering name and email id you will get error message show in the below figure...

This screenshot shows the same form as above, but with error messages. The 'Your Name:' and 'Your E-mail:' fields now have red text '* required' next to them, indicating that these fields are mandatory. The other fields and the 'Submit' button remain the same.

Entering all the details shown in the following diagram and click on submit button

The screenshot shows a web browser window with the address bar displaying 'localhost/sakshipost/index.php'. The page title is 'PHP Form Validation Example'. The form contains the following elements:

- 'Your Name:' followed by a text input field containing 'sakshi' and a red asterisk indicating a required field.
- 'Your E-mail:' followed by a text input field containing 'career@sakshieducation.com' and a red asterisk.
- 'Your Age:' followed by a text input field containing '25'.
- 'Gender:' followed by two radio buttons: 'Female' (unselected) and 'Male' (selected).
- A text area containing the text 'i am searching for job'.
- 'Message : ' label to the left of the text area.
- A 'Submit' button at the bottom left.

After entering all the required details, click the submit button then you get the output as shown in below figure.

Output

localhost/sakshipost/index.php

PHP Form Validation Example

Your Name: *

Your E-mail: *

Your Age:

Gender: Female Male

Message :

Details you entered:

sakshi
career@sakshieducation.com
25
male
i am searching for job

PHP - Required Fields

In the above program, we have Name and Email Id fields and remaining fields are optional. By using PHP we can define required fields.

Example

```
<!DOCTYPE HTML>
<html>
<head>
</head>
<body>
<?php
// define and initialize variables and set them to empty
$name_err = "";
$email_err = "";
$gender_err = "";
```

```
$age_err = "";
$name = $email = $gender = $comment = $age = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $name_err = "Your Name is required";
    } else {
        $name = $_POST["name"];
    }

    if (empty($_POST["email"])) {
        $email_err = "Email is required";
    } else {
        $email = $_POST["email"];
    }

    if (empty($_POST["age"])) {
        $age = "";
    } else {
        $age = $_POST["age"];
    }

    if (empty($_POST["message"])) {
        $message = "";
    } else {
        $message = $_POST["message"];
    }

    if (empty($_POST["gender"])) {
        $gender = "";
    } else {
        $gender = $_POST["gender"];
    }
}
?>
<h2>PHP Form Validation Example</h2>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
Your Name: <input type="text" name="name"> <span style="color:red;">* <?php echo $name_err;
?></span><br/><br/>
Your E-mail: <input type="text" name="email"> <span style="color:red;">* <?php echo $email_err;
?></span><br/><br/>
Your Age: <input type="text" name="age"><br/><br/>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male<br/><br/>
Message : <textarea name="message" rows="5" cols="40"></textarea><br/><br/>
<input type="submit" name="submit" value="Submit">
</form>

<?php
if($name != "" || $email != "") {
```



```
echo "<h2>Details you entered:</h2>";  
echo "<b>you are name</b>".$name;  
echo "<br />";  
echo "<b>your email id</b>".$email;  
echo "<br />";  
echo "<b>your age </b>".$age;  
echo "<br />";  
echo "<b>you are </b>".$gender;  
echo "<br />";  
echo "<b>you entered </b>".$message;  
}  
?>  
</body>  
</html>
```

Output

localhost/sakshipost/index.php

PHP Form Validation Example

Your Name: *

Your E-mail: *

Your Age:

Gender: Female Male

Message :

Details you entered:

you are name sakshi
your email id info@sakshieducation.com
your age 24
you are male
you entered Hay ! This is SakshiEducation

If you enter Name as Sakshi, email id as *info@sakshieducation.com*, age as 24, gender as male and message as "Hay !This is SakshiEducation", then the output will be shown in the above example.

Email-validation

Regular expressions are nothing but a sequence or pattern of characters. They give us the strength of pattern-matching functionality.

For Email validation or for Name validation or for Url validation or for any pattern validation we use Regular expression in PHP.

```
<!DOCTYPE HTML>
<html>
<head>
  <style>
    .error {color: red;}
  </style>
</head>
<body>

<?php
// define variables and set to empty values
$name_err = "";
$email_err = "";
$gender_err = "";
$website_err = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    $name_err = "Enter your name here";
  } else {
    $name = $_POST["name"];
    // check if name only contains letters and whitespace
    if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
      $name_err = "Please enter letters and spaces";
    }
  }

  if (empty($_POST["email"])) {
    $email_err = "Email is required";
  } else {
    $email = $_POST["email"];
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
      $email_err = "Enter valid Email format";
    }
  }

  if (empty($_POST["message"])) {
    $message = "";
  } else {
    $message = $_POST["message"];
  }

  if (empty($_POST["gender"])) {
    $gender = "";
  }
}
```

```
    } else {
        $gender = $_POST["gender"];
    }
}

?>

<h2>Email validation with regular expressions</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    Name: <input type="text" name="name">
    <span class="error">* <?php echo $name_err;?></span>
    <br><br>
    E-mail: <input type="text" name="email">
    <span class="error">* <?php echo $email_err;?></span>
    <br><br>
    Message: <textarea name="message" rows="5" cols="40"
style="vertical-align: middle;"></textarea>
    <br><br>
    Gender:
    <input type="radio" name="gender" value="female">Female
    <input type="radio" name="gender" value="male">Male

    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>

<?php
if(empty($email_err) && empty($name_err)) {
    echo "<h3>Details you entered:</h3>";
    echo $name;
    echo "<br>";
    echo $email;
    echo "<br>";
    echo $message;
    echo "<br>";
    echo $gender;
}
?>
</body>
</html>
```

Output



localhost/sakshipost/index.php

Email validation with regular expressions

Name: *

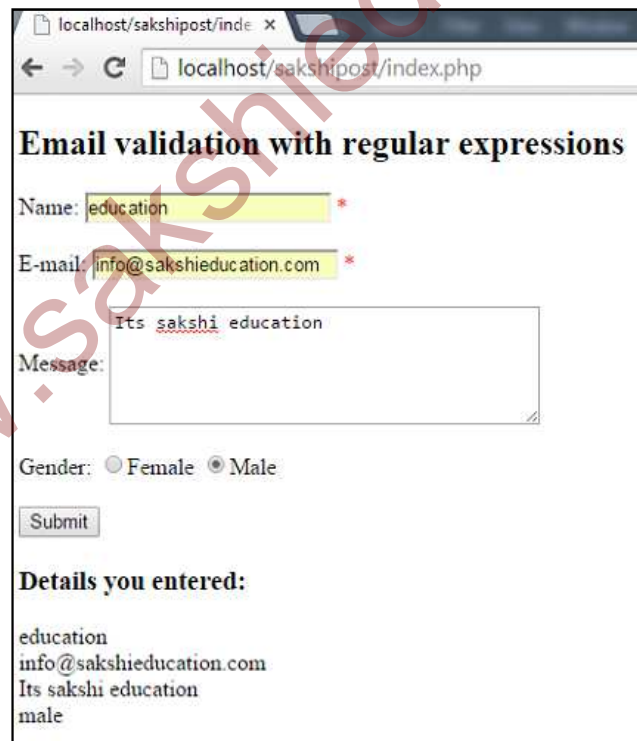
E-mail: *

Message:

Gender: Female Male

In above example Name and Email id are required fields and Name will take characters and spaces where as email will be validated only with mail format. Ex: info@sakshieducation.com.

If you enter wrongly it will ask you to enter valid details.



localhost/sakshipost/index.php

Email validation with regular expressions

Name: education *

E-mail: info@sakshieducation.com *

Message: Its sakshi education

Gender: Female Male

Details you entered:

education
info@sakshieducation.com
Its sakshi education
male

If you enter all the details correctly then you will get the output as follows:

To validate URL that is a website link then use the following code as like email

The screenshot shows a web browser window with the address bar displaying 'localhost/sakshipost/index.php'. The page title is 'Email validation with regular expressions'. The form contains the following fields and values:

- Name: education
- E-mail: info@sakshieducation.com
- Message: Its sakshi education
- Gender: Female Male

A 'Submit' button is located below the form. Under the heading 'Details you entered:', the following information is displayed:

```
education
info@sakshieducation.com
Its sakshi education
male
```

```
$website = $_POST["website"]); if (!preg_match("/\b(?:?:https?|ftp):\\V\\|www\\.)[-a-z0-9+&@#\\/%?~_!|:,;]*[-a-z0-9+&@#\\/%=~_]|/i",$website) { $website_err = "Invalid Website/url";}
```

If you want to place the values in the form after submitting then use the following program.

```
<!DOCTYPE HTML>
<html>
<head>
  <style>
    .error {color: red;}
  </style>
</head>
<body>
<?php
// define variables and set to empty values
$name_err = "";
$email_err = "";
$gender_err = "";
$website_err = "";
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
if (empty($_POST["name"])) {
    $name_err = "Enter your name here";
} else {
    $name = $_POST["name"];
    // check if name only contains letters and whitespace
    if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
        $name_err = "Please enter letters and spaces";
    }
}

if (empty($_POST["email"])) {
    $email_err = "Email is required";
} else {
    $email = $_POST["email"];
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $email_err = "Enter valid Email format";
    }
}

if (empty($_POST["message"])) {
    $message = "";
} else {
    $message = $_POST["message"];
}

if (empty($_POST["gender"])) {
    $gender = "";
} else {
    $gender = $_POST["gender"];
}
}

?>

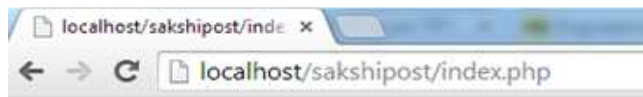
<h2>Keep the values in the form</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    Name: <input type="text" name="name" value="<?php echo $name;?>">
    <span class="error">* <?php echo $name_err;?></span>
    <br><br>
    E-mail: <input type="text" name="email" value="<?php echo
$email;?>">
    <span class="error">* <?php echo $email_err;?></span>
    <br><br>
    Message: <textarea name="message" rows="5" cols="40"
style="vertical-align: middle;" >
        <?php echo $message;?>
    </textarea>
    <br><br>
    Gender:
    <input type="radio" name="gender" <?php if (isset($gender) &&
$gender=="female") echo "checked";?> value="female">Female
    <input type="radio" name="gender" <?php if (isset($gender) &&
$gender=="male") echo "checked";?> value="male">Male
```

```
<br><br>
<input type="submit" name="submit" value="Submit">
</form>

<?php
if(empty($email_err) && empty($name_err)) {
    echo "<h3>Details you entered:</h3>";
    echo $name;
    echo "<br>";
    echo $email;
    echo "<br>";
    echo $message;
    echo "<br>";
    echo $gender;
}
?>

</body>
</html>
```

Output



Keep the values in the form

Name: *

E-mail: *

Message:

Gender: Female Male

Details you entered:

sakshi
info@sakshieducation.com
This is sakshi
male