PHP Form Processing

Last Updated : 15 Apr, 2020

In this article, we will discuss how to process form in PHP. HTML forms are used to send the user information to the server and returns the result back to the browser. For example, if you want to get the details of visitors to your website, and send them good thoughts, you can collect the user information by means of form processing. Then, the information can be validated either at the client-side or on the server-side. The final result is sent to the client through the respective web browser. To create a HTML form, **form** tag should be used.

Attributes of Form Tag:

Attribute	Description
name or id	It specifies the name of the form and is used to identify individual forms.
action	It specifies the location to which the form data has to be sent when the form is submitted.
method	It specifies the HTTP method that is to be used when the form is submitted. The possible values are **get** and **post**. If **get** method is used, the form data are visible to the users in the url. Default HTTP method is **get**.
encType	It specifies the encryption type for the form data when the form is submitted.
novalidate	It implies the server not to verify the form data when the form is submitted.

Controls used in forms: Form processing contains a set of controls through which the client and server can communicate and share information. The controls used in forms are:

- **Textbox:** Textbox allows the user to provide single-line input, which can be used for getting values such as names, search menu and etc.
- **Textarea:** Textarea allows the user to provide multi-line input, which can be used for getting values such as an address, message etc.
- **DropDown:** Dropdown or combobox allows the user to provide select a value from a list of values.
- **Radio Buttons:** Radio buttons allow the user to select only one option from the given set of options.
- **CheckBox:** Checkbox allows the user to select multiple options from the set of given options.
- **Buttons:** Buttons are the clickable controls that can be used to submit the form.

Creating a simple HTML Form: All the form controls given above is designed by using the **input** tag based on the **type** attribute of the tag. In the below script, when the form is submitted, no event handling mechanism is done. Event handling refers to the process done while the form is submitted. These event handling mechanisms can be done by using javaScript or PHP. However, JavaScript provides only client-side validation. Hence, we can use PHP for form processing.

HTML Code:

```
<!DOCTYPE html>
<html>

<head>
    <title>Simple Form Processing</title>
</head>

<body>
    <form id="form1" method="post">
        FirstName:
        <input type="text" name="firstname" required/>
        <br>
        <br>
        LastName
        <input type="text" name="lastname" required/>
        <br>
        <br>
        Address
        <input type="text" name="address" required/>
        <br>
        <br>
        Email Address:
        <input type="email" name="emailaddress" required/>
        <br>
        <br>
        Password:
        <input type="password" name="password" required/>
        <br>
        <br>
        <input type="submit" value="Submit”/>
    </form>
</body>
</html>
```

Form Validation: Form validation is done to ensure that the user has provided the relevant information. Basic validation can be done using HTML elements. For example, in the above script, the email address text box is having a type value as “email”, which prevents the user from entering the incorrect value for an email. Every form field in the above script is followed by a required attribute, which will intimate the user not to leave any field empty before submitting the form. PHP methods and arrays used in form processing are:

- **isset():** This function is used to determine whether the variable or a form control is having a value or not.
- **$_GET[]:** It is used the retrieve the information from the form control through the parameters sent in the URL. It takes the attribute given in the url as the parameter.

A HTML form submits information via the HTTP POST method if the form's method attribute is set to "POST"

- **$_POST[]:** It is used the retrieve the information from the form control through the HTTP POST method. IT takes name attribute of corresponding form control as the parameter.
- **$_REQUEST[]:** It is used to retrieve an information while using a database.

Form Processing using PHP: Above HTML script is rewritten using the above mentioned functions and array. The rewritten script validates all the form fields and if there are no errors, it displays the received information in a tabular form.

- **Example:**

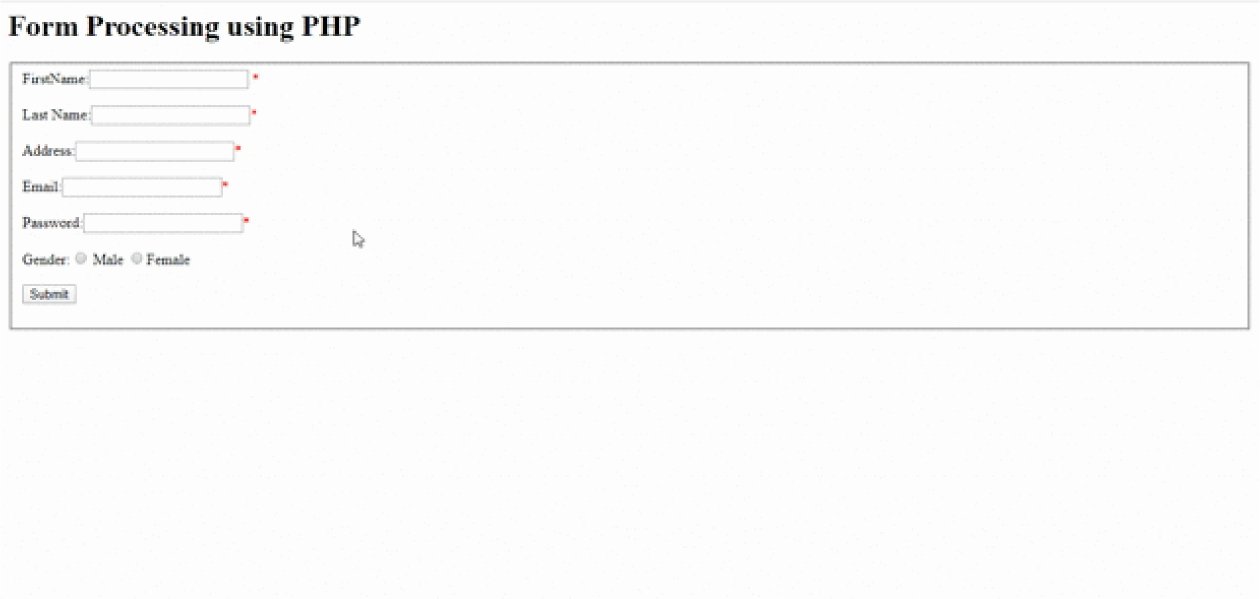
```
<?php
```

```
if (isset($_POST['submit']))
{
    if ((!isset($_POST['firstname'])) || (!isset($_POST['lastname'])) ||
        (!isset($_POST['address'])) || (!isset($_POST['emailaddress'])) ||
        (!isset($_POST['password'])) || (!isset($_POST['gender'])))
    {
        $error = "*" . "Please fill all the required fields";
    }
    else
    {
        $firstname = $_POST['firstname'];
        $lastname = $_POST['lastname'];
        $address = $_POST['address'];
        $emailaddress = $_POST['emailaddress'];
        $password = $_POST['password'];
        $gender = $_POST['gender'];
    }
}
?>
<html>

<head>
    <title>Simple Form Processing</title>
</head>

<body>
    <h1>Form Processing using PHP</h1>
    <fieldset>
        <form id="form1" method="post" action="form.php">
            <?php
                if (isset($_POST['submit']))
                {
                    if (isset($error))
                    {
                        echo "<p style='color:red;'>"
                        . $error . "</p>";
                    }
                }
                ?>

                FirstName:
                <input type="text" name="firstname"/>
                 <span style="color:red;">*</span>
                <br>
                <br>
                Last Name:
                <input type="text" name="lastname"/>
```

```
              <span style="color:red;">*</span>
            <br>
            <br>
            Address:
            <input type="text" name="address"/>
              <span style="color:red;">*</span>
            <br>
            <br>
            Email:
            <input type="email" name="emailaddress"/>
              <span style="color:red;">*</span>
            <br>
            <br>
            Password:
            <input type="password" name="password"/>
               <span style="color:red;">*</span>
            <br>
            <br>
            Gender:
            <input type="radio"
                   value="Male"
                   name="gender"> Male
            <input type="radio"
                   value="Female"
                   name="gender">Female
            <br>
            <br>
            <input type="submit" value="Submit" name="submit" />
    </form>
</fieldset>
<?php
   if(isset($_POST['submit']))
   {
      if(!isset($error))
       {
               echo"<h1>INPUT RECEIVED</h1><br>";
               echo "<table border='1'>";
               echo "<thead>";
               echo "<th>Parameter</th>";
               echo "<th>Value</th>";
               echo "</thead>";
               echo "<tr>";
               echo "<td>First Name</td>";
               echo "<td>".$firstname."</td>";
               echo "</tr>";
               echo "<tr>";
               echo "<td>Last Name</td>";
```

```
                  echo "<td>".$lastname."</td>";
                  echo "</tr>";
                  echo "<tr>";
                  echo "<td>Address</td>";
                  echo "<td>".$address."</td>";
                  echo "</tr>";
                  echo "<tr>";
                  echo "<td>Email Address</td>";
                  echo "<td>" .$emailaddress."</td>";
                  echo "</tr>";
                  echo "<tr>";
                  echo "<td>Password</td>";
                  echo "<td>".$password."</td>";
                  echo "</tr>";
                  echo "<tr>";
                  echo "<td>Gender</td>";
                  echo "<td>".$gender."</td>";
                  echo "</tr>";
                  echo "</table>";
        }
      }
    ?>
</body>

</html>
```

- **Output:**



Form Processing using PHP

FirstName: *

Last Name: *

Address: *

Email: *

Password: *

Gender: ○ Male ○ Female

Note: When the PHP and HTML are coded in a single file, the file should be saved as PHP. In the form, the value for the action parameter should be a file name.

PHP Form Processing

Comment

More info

Advertise with us

Next Article

PHP Date and Time

Similar Reads

PHP Tutorial

PHP (Hypertext Preprocessor) is a versatile and widely used server-side scripting language for creating dynamic and interactive web applications. This PHP tutorial will give you an in-depth understanding of...

6 min read

PHP Introduction

The term PHP is an acronym for - Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. It is an open-source which means it is free to download and...

8 min read

PHP Basics

PHP Array

PHP Constants

PHP Operators

PHP Control Statements

PHP Loops

PHP Functions

PHP Advanced

PHP | Superglobals

We already have discussed about variables and global variables in PHP in the post PHP | Variables and Data Types. In this article, we will learn about superglobals in PHP. These are specially-defined array...

🕓 5 min read

HTTP GET and POST Methods in PHP

In this article, we will know what HTTP GET and POST methods are in PHP, how to implement these HTTP methods & their usage, by understanding them through the examples. HTTP: The Hypertext...

🕓 4 min read

PHP | Regular Expressions

Regular expressions commonly known as a regex (regexes) are a sequence of characters describing a special search pattern in the form of text string. They are basically used in programming world...

🕓 11 min read

PHP Form Processing

In this article, we will discuss how to process form in PHP. HTML forms are used to send the user information to the server and returns the result back to the browser. For example, if you want to get th...

🕓 5 min read

PHP Date and Time

In this article, we will see how to get the date & time using the date() & time() function in PHP, we will also see the various formatting options available with these functions & understand their...

🕓 4 min read

Describe PHP Include and Require

In this article, we will see what include() & the require() functions is, also will know how these functions affect the execution of the code, their differences & usage in PHP, along with understanding their...

🕓 3 min read

PHP File Handling

File handling in PHP is used to you to create, open, read, write, delete, and manipulate files on a server. It is used when you need to store data persistently or handle files uploaded by users. PHP provides...

🕓 3 min read

PHP | Uploading File

Have you ever wondered how websites build their system of file uploading in PHP? Here we will come to know about the file uploading process. A question which you can come up with - 'Are we able to...

3 min read

PHP Cookies

A cookie in PHP is a small file with a maximum size of 4KB that the web server stores on the client computer. They are typically used to keep track of information such as a username that the site can...

5 min read

PHP | Sessions

What is a session? In general, session refers to a frame of communication between two medium. A PHP session is used to store data on a server rather than the computer of the user. Session identifiers or SID...

3 min read

Implementing callback in PHP

In PHP, callback is a function object/reference with type callable. A callback/callable variable can act as a function, object method and a static class method. There are various ways to implement a callback....

3 min read