

PHP - Form Handling

HTML Forms play an important role in PHP web applications. Although a webpage composed purely with HTML is a static webpage, the HTML form component is an important feature that helps in bringing interactivity and rendering dynamic content. PHP's form handling functionality can validate data collected from the user, before processing.

An HTML Form is a collection various form controls such as text fields, checkboxes, radio buttons, etc., with which the user can interact, enter or choose certain data that may be either locally processed by JavaScript (client-side processing), or sent to a remote server for processing with the help of server-side programming scripts such as PHP.

One or more form control elements are put inside `<form>` and `</form>` tags. The **form element** is characterized by different attributes such as name, action, and method.

```
<form [attributes]>  
    Form controls  
</form>
```

Form Attributes

Out of the many attributes of the HTML form element, the following attributes are often required and defined –

Action Attribute

a string representing the URL that processes the form submission. For example, **`http://example.com/test.php`**. To submit the form-data to the same PHP script in which the HTML form is defined, use the `PHP_SELF` server variable –

```
<form action="php echo $_SERVER['PHP_SELF']; ?" method="post">
```

Enctype Attribute

specifies the method using which the form-data should be encoded before sending it to the server. Possible values are –

- **application/x-www-form-urlencoded** – The default value.
- **multipart/form-data** – Use this if the form contains `<input>` elements with `type=file`.

- **text/plain** – Useful for debugging purposes.

Method Attribute

a string representing the HTTP method to submit the form with. The following methods are the possible values of method attribute –

- **post** – The POST method; form data sent as the request body.
- **get (default)** – The GET; form data appended to the action URL with a "?" separator. Use this method when the form has no side effects.
- **dialog** – When the form is inside a <dialog>, closes the dialog and causes a submit event to be fired on submission, without submitting data or clearing the form.

Name Attribute

The name of the form. The value must not be the empty string, and must be unique if there are multiple forms in the same HTML document.

Target Attribute

a string that indicates where to display the response after submitting the form. Should be one of the following –

- **_self (default)** – Load into the same browsing context as the current one.
- **_blank** – Load into a new unnamed browsing context.
- **_parent** – Load into the parent browsing context of the current one.
- **_top** – Load into the top-level browsing context (an ancestor of the current one and has no parent).

Hence, a typical HTML form, used in a PHP web application looks like –

```
<form name="form1" action="php echo $_SERVER['PHP_SELF'];?&gt;" action="POST"&gt;<br/    Form controls  
</form>
```

Form Elements

A HTML form is designed with different types of controls or elements. The user can interact with these controls to enter data or choose from the available options presented. Some of the elements are described below –

Input Element

The input element represents a data field, which enables the user to enter and/or edit the data.

The type attribute of INPUT element controls the data. The INPUT element may be of the following types –

Text

A text field to enter a single line text.

```
<input type="text" name="employee">
```

Password

A single line text field that masks the entered characters.

```
<input type="password" name="pwd"><br>
```

Checkbox

A rectangular checkable box which is a set of zero or more values from a predefined list.

```
<input type="checkbox" id="s1" name="sport1" value="Cricket">  
<label for="s1">I like Cricket</label><br>  
<input type="checkbox" id="s2" name="sport2" value="Football">  
<label for="s2">I like Football</label><br>  
<input type="checkbox" id="s3" name="sport3" value="Tennis">  
<label for="s3">I like Tennis</label><br><br>
```

Radio

This type renders a round clickable button with two states (ON or OFF), usually a part of multiple buttons in a radio group.

```
<input type="radio" id="g1" name="gender" value="Male">  
<label for="g1">Male</label><br>
```

```
<input type="radio" id="g2" name="female" value="Female">
<label for="g2">Female</label><br>
```

File

The input type renders a button captioned file and allows the user to select a file from the client filesystem, usually to be uploaded on the server. The form's enctype attribute must be set to "multipart/form-data"

```
<input type="file" name="file">
```

Email

A single line text field, customized to accept a string conforming to valid email ID.

URL

A single line text field customized to accept a string conforming to valid URL.

Submit

This input element renders a button, which when clicked, initiates the the submission of form data to the URL specified in the action attribute of the current form.

```
<input type="submit" name="Submit">
```

Select Element

The select element represents a control for selecting amongst a set of options. Each choice is defined with option attribute of Select Control. For example –

```
<select name="Subjects" id="subject">
  <option value="Physics">Physics</option>
  <option value="Chemistry">Chemistry</option>
  <option value="Maths">Maths</option>
  <option value="English">English</option>
</select>
```

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Form Example

Let us use these form elements to design a HTML form and send it to a PHP_SELF script

```
<html>
<body>
  <form method = "post" action = "<?php
    echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    <table>
      <tr>
        <td>Name:</td>
        <td><input type = "text" name = "name"></td>
      </tr>
      <tr>
        <td>E-mail: </td>
        <td><input type = "email" name = "email"></td>
      </tr>
      <tr>
        <td>Website:</td>
        <td><input type = "url" name = "website"></td>
      </tr>
      <tr>
        <td>Classes:</td>
        <td><textarea name = "comment" rows = "5" cols = "40"></textarea></td>
      </tr>
      <tr>
        <td>Gender:</td>
        <td>
          <input type = "radio" name = "gender" value = "female">Female
          <input type = "radio" name = "gender" value = "male">Male
        </td>
      </tr>
      <td>
        <input type = "submit" name = "submit" value = "Submit">
      </td>
    </table>
  </form>
  <?php
    $name = $email = $gender = $comment = $site = "";

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      $name = $_POST["name"];
```

```
$email = $_POST["email"];
$name = $_POST["name"];
$comment = $_POST["comment"];
$gender = $_POST["gender"];
$site = $_POST["website"];
}
echo "<h2>Your given values are as:</h2>";
echo $name;
echo "<br>";

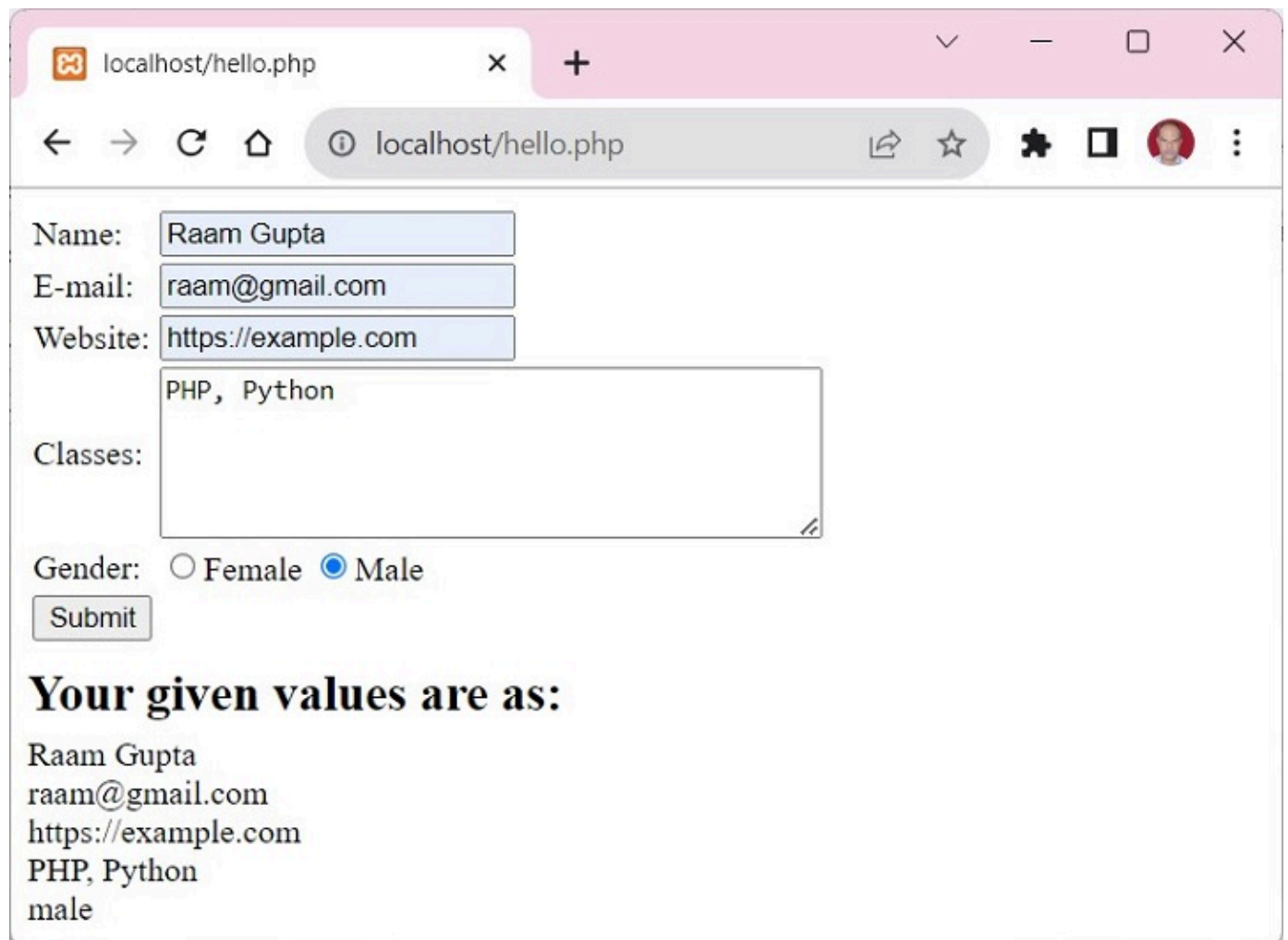
echo $email;
echo "<br>";

echo $site;
echo "<br>";

echo $comment;
echo "<br>";

echo $gender;
?>
</body>
</html>
```

It will produce the following **output** –



The screenshot shows a web browser window with the address bar displaying 'localhost/hello.php'. The page contains a form with the following fields and values:

- Name: Raam Gupta
- E-mail: raam@gmail.com
- Website: https://example.com
- Classes: PHP, Python
- Gender: ☐ Female ☒ Male

Below the form is a 'Submit' button. Underneath the button, the text 'Your given values are as:' is displayed, followed by the values entered in the form:

Raam Gupta
raam@gmail.com
https://example.com
PHP, Python
male