MODUL PRAKTIKUM

PEMROGRAMAN WEB

Materi 13:
PHP FORM

Dosen:

Febrianta Surya Nugraha, M.Kom

S1 Informatika

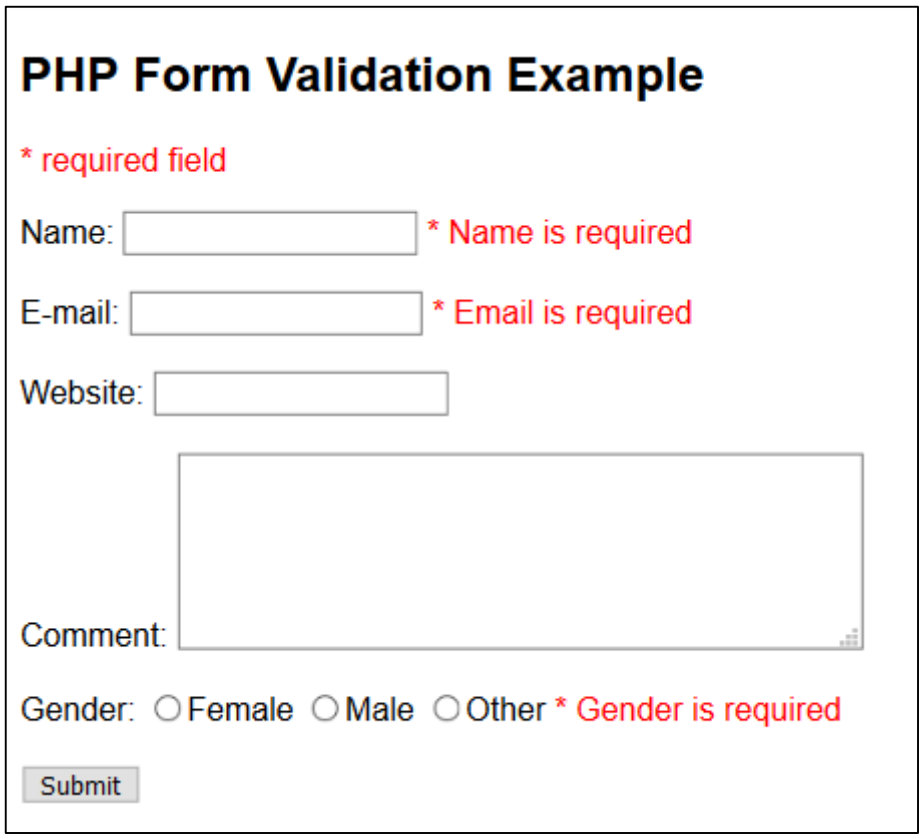
STMIK AMIKOM SURAKARTA

2020

PHP Form

PHP Form Validation

HTML form yang akan kita bahas dalam bab-bab ini, berisi berbagai bidang input: required and optional text fields, radio buttons, and a submit button:



Aturan validasi untuk form di atas adalah sebagai berikut:

Field	***Validation Rules***
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

Pertama kita akan melihat kode HTML biasa untuk form:

Text Fields

Field Name, email, and website adalah element text input, and the comment field adalah textarea. Kode HTML terlihat seperti ini:

```
Name: <input type="text" name="name">
E-mail: <input type="text" name="email">
Website: <input type="text" name="website">
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
```

Radio Buttons

Gender fields adalah radio button, Kode HTML:

```
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
```

The Form Element

HTML code dari form:

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

Ketika form disubmit, data form dikirim dengan method="post".

Apakah variable $_SERVER["PHP_SELF"]?
The $_SERVER["PHP_SELF"] adalah variabel super global yang mengembalikan filename skrip yang saat ini sedang dijalankan.

Jadi, $ _SERVER ["PHP_SELF"] mengirimkan data formulir yang dikirimkan ke halaman itu sendiri, daripada melompat ke halaman lain. Dengan cara ini, pengguna akan mendapatkan pesan kesalahan pada halaman yang sama dengan formulir.

Apakakah htmlspecialchars() function?
The htmlspecialchars() function mengkonversi special characters menjadi entitas HTML. Ini berarti akan menggantikan karakter HTML seperti < dan > menjadi < dan >. Ini mencegah penyerang mengeksploitasi kode dengan menyuntikkan kode HTML atau Javascript (Cross-site Scripting attacks) di form.

Big Note on PHP Form Security

The $_SERVER["PHP_SELF"] variable dapat digunakan oleh hackers!

Jika PHP_SELF digunakan di halaman Anda maka pengguna dapat memasukkan slash (/) dan kemudian beberapa perintah Cross Site Scripting (XSS) untuk mengeksekusi.

Cross-site scripting (XSS) adalah jenis kerentanan keamanan komputer yang biasanya ditemukan di aplikasi Web. XSS memungkinkan penyerang untuk menyuntikkan skrip sisi klien ke halaman Web yang dilihat oleh pengguna lain.

Asumsikan kita memiliki form berikut di halaman bernama "test_form.php":

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

Sekarang, jika pengguna memasukkan URL normal di address bar seperti "http://www.example.com/test_form.php", kode di atas akan diterjemahkan ke:

```
<form method="post" action="test_form.php">
```

Sejauh ini baik. Namun, pertimbangkan bahwa pengguna memasukkan URL berikut di address bar:

```
http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E
```

Dalam hal ini, kode di atas akan diterjemahkan ke:

```
<form method="post" action="test_form.php/"><script>alert('hacked')</script>
```

Kode ini menambahkan tag script dan perintah alert. Dan ketika halaman dimuat, kode JavaScript akan dieksekusi (pengguna akan melihat kotak peringatan). Ini hanyalah contoh sederhana dan tidak berbahaya bagaimana variabel PHP_SELF dapat dieksploitasi.

Ketahuilah bahwa kode JavaScript apa pun dapat ditambahkan di dalam tag <script>! Hacker misalnya dapat mengarahkan pengguna ke file di server lain, dan file itu dapat menyimpan kode jahat yang dapat mengubah variabel global atau mengirimkan formulir ke alamat lain untuk menyimpan data pengguna.

How To Avoid $_SERVER["PHP_SELF"] Exploits?

Eksploitasi $ _SERVER ["PHP_SELF"] dapat dihindari dengan menggunakan fungsi htmlspecialchars (). Form code:

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

Fungsi htmlspecialchars () mengubah karakter khusus menjadi entitas HTML. Sekarang jika pengguna mencoba untuk mengeksploitasi variabel PHP_SELF, itu akan menghasilkan output berikut:

```
<form method="post"
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

Upaya eksploitasi gagal dan tidak ada kerusakan yang terjadi.

Validate Form Data With PHP

Hal pertama yang akan kita lakukan adalah meneruskan semua variabel melalui fungsi PHP htmlspecialchars ().

Saat menggunakan fungsi htmlspecialchars (); lalu jika pengguna mencoba mengirimkan hal berikut dalam bidang teks::

```
<script>location.href('http://www.hacked.com')</script>
```

- ini tidak akan dieksekusi, karena akan disimpan sebagai kode escape HTML, seperti ini:

```
&lt;script&gt;location.href('http://www.hacked.com')&lt;/script&gt;
```

Kode sekarang aman untuk ditampilkan pada halaman atau di dalam e-mail.

Kita juga akan melakukan dua hal lagi ketika pengguna mengirimkan formulir:

1. Hilangkan karakter yang tidak perlu (extra space, tab, newline) dari data input pengguna (dengan fungsi PHP trim ())
2. Hapus backslash (\) dari data input pengguna (dengan fungsi PHP stripslashes ())

Langkah selanjutnya adalah membuat fungsi yang akan melakukan semua pemeriksaan untuk kita (yang jauh lebih nyaman daripada menulis kode yang sama berulang-ulang).

Kami akan memberi nama fungsi test_input ().

Sekarang, kita dapat memeriksa setiap variabel $ _POST dengan fungsi test_input (), dan skripnya terlihat seperti ini:

```php
<!DOCTYPE HTML>
<html>
<head>
</head>
<body>

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

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $name = test_input($_POST["name"]);
  $email = test_input($_POST["email"]);
  $website = test_input($_POST["website"]);
  $comment = test_input($_POST["comment"]);
  $gender = test_input($_POST["gender"]);
}

function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>

<h2>PHP Form Validation Example</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  Name: <input type="text" name="name">
  <br><br>
  E-mail: <input type="text" name="email">
  <br><br>
  Website: <input type="text" name="website">
  <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
  <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>
```

Perhatikan bahwa pada awal skrip, akan memeriksa apakah formulir telah dikirimkan menggunakan $ _SERVER ["REQUEST_METHOD"]. Jika REQUEST_METHOD adalah POST, maka formulir telah dikirimkan - dan itu harus divalidasi. Jika belum dikirim, lewati validasi dan tampilkan formulir kosong.

Namun, dalam contoh di atas, semua bidang input adalah opsional. Script berfungsi dengan baik bahkan jika pengguna tidak memasukkan data apa pun. Langkah selanjutnya adalah membuat kolom input yang diperlukan dan membuat pesan kesalahan jika diperlukan.

PHP - REQUIRED FIELDS

Dari tabel aturan validasi pada halaman sebelumnya, kita melihat bahwa field Name", "E-mail", dan "Gender" diperlukan. Field ini tidak boleh kosong dan harus diisi dalam form HTML

Field	***Validation Rules***
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

Pada bab sebelumnya, semua input field adalah opsional.

Dalam kode berikut ini telah menambahkan beberapa variabel baru: $nameErr, $emailErr, $genderErr, dan $websiteErr. Variabel kesalahan ini akan menyimpan pesan kesalahan untuk field yang diperlukan. Kita juga telah menambahkan pernyataan if else untuk setiap variabel $_POST. Ini memeriksa apakah variabel $_POST kosong (dengan fungsi PHP empty ()). Jika kosong, pesan kesalahan disimpan dalam variabel kesalahan yang berbeda, dan jika tidak kosong, ia mengirimkan data input pengguna melalui fungsi test_input ():

```php
<?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"]);
  }

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

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

  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"]);
  }
}
?>
```

PHP - Display The Error Messages

Kemudian dalam form HTML, ditambahkan skrip kecil setelah setiap field yang diperlukan, yang menghasilkan pesan kesalahan yang benar jika diperlukan (yaitu jika pengguna mencoba mengirimkan form tanpa mengisi required fields):

```
<!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"]);
  }

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

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

  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;
}
```

Script lanjut ke halaman berikutnya. . .

```
?>

<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>
```

Langkah selanjutnya adalah memvalidasi data input, yaitu "Apakah field Nama hanya berisi huruf dan spasi?", Dan "Apakah field Email berisi sintaks alamat email yang valid?", Dan jika diisi, " Apakah field Website berisi URL yang valid? ".

PHP FORMS - VALIDATE E-MAIL AND URL

PHP - Validate Name

Kode di bawah ini menunjukkan cara sederhana untuk memeriksa apakah field name hanya berisi huruf dan spasi. Jika nilai field name tidak valid, maka menyimpan pesan kesalahan:

```
$name = test_input($_POST["name"]);
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
  $nameErr = "Only letters and white space allowed";
}
```

Fungsi preg_match() mencari string untuk pola, mengembalikan true jika polanya ada, dan false jika tidak.

PHP - Validate E-mail

Cara termudah dan teraman untuk memeriksa apakah alamat email terbentuk dengan baik adalah dengan menggunakan fungsi filter_var () PHP. Dalam kode di bawah ini, jika alamat email tidak terbentuk dengan baik, maka menyimpan pesan kesalahan:

```
$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
  $emailErr = "Invalid email format";
}
```

PHP - Validate URL

Kode di bawah ini menunjukkan cara untuk memeriksa apakah sintaksis alamat URL valid (ekspresi reguler ini juga memungkinkan tanda hubung di URL). Jika sintaks alamat URL tidak valid, maka menyimpan pesan kesalahan:

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

PHP - Validate Name, E-mail, and URL

Sekarang, skripnya terlihat seperti ini:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
```

```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 (!filter_var($email, FILTER_VALIDATE_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>
```

PHP COMPLETE FORM EXAMPLE

PHP - Keep The Values in The Form

Untuk menampilkan nilai di field input setelah pengguna menekan tombol submit, ditambahkan skrip PHP kecil di dalam atribut nilai field input berikut: nama, email, dan website. Di bidang textarea comment, ditempatkan skrip di antara tag <textarea> dan </textarea>. Skrip kecil menampilkan nilai variabel $name, $email, $website, and $comment variables.

Kemudian, kita juga perlu menunjukkan tombol radio mana yang diperiksa. Untuk ini, kita harus memanipulasi atribut yang diperiksa (bukan atribut nilai untuk tombol radio):

```
Name: <input type="text" name="name" value="<?php echo $name;?>">

E-mail: <input type="text" name="email" value="<?php echo $email;?>">

Website: <input type="text" name="website" value="<?php echo $website;?>">

Comment: <textarea name="comment" rows="5" cols="40"><?php echo $comment;?></textarea>

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) && $gender=="male") echo "checked";?>
value="male">Male
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="other") echo "checked";?>
value="other">Other
```

PHP - Complete Form Example

Berikut adalah kode lengkap untuk Contoh Validasi Formulir PHP:

```
<!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 (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
```

```
    $emailErr = "Invalid email format";
   }
  }

  if (empty($_POST["website"])) {
   $website = "";
  } else {
   $website = test_input($_POST["website"]);
   // check if URL address syntax is valid (this regular expression also allows dashes in the URL)
   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" value="<?php echo $name;?>">
  <span class="error">* <?php echo $nameErr;?></span>
  <br><br>
  E-mail: <input type="text" name="email" value="<?php echo $email;?>">
  <span class="error">* <?php echo $emailErr;?></span>
  <br><br>
  Website: <input type="text" name="website" value="<?php echo $website;?>">
  <span class="error"><?php echo $websiteErr;?></span>
  <br><br>
  Comment: <textarea name="comment" rows="5" cols="40"><?php echo $comment;?></textarea>
  <br><br>
  Gender:
  <input type="radio" name="gender" <?php if (isset($gender) && $gender=="female") echo "checked";?>
value="female">Female
  <input type="radio" name="gender" <?php if (isset($gender) && $gender=="male") echo "checked";?>
value="male">Male
  <input type="radio" name="gender" <?php if (isset($gender) && $gender=="other") echo "checked";?>
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>
```

PHP Form Validation Example

* required field

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male Other *

Submit

Your Input: