

PHP - Form Validation

The term "Form Validation" refers to the process of ascertaining if the data entered by the user in various form elements is acceptable for further processing. Validation of data before its subsequent processing avoids possible exceptions and runtime errors.

Validation can be done both on the client-side and on the server-side. When the client submits the form, the form data is intercepted by the PHP script running on the server. Using various functions available in PHP, the server-side form validation can be done.

Client-side Validation

The new input controls as per the HTML5 specifications have in-built validation. For example an input element of the type 'email', even though is a text field, is customized to accept a string that is according to email address protocol.

Validation takes place before the data is submitted to the server. Same thing is true with other input types such as URL, number, etc.

Example

Given below is an HTML form with input elements of number type, email type and URL type. If you enter data that is not as per the required format, a suitable error message is flashed as you try to submit the form.

</>

Open Compiler

```
<h1>Input Validation</h1>
<form>
  <p><Label for "name">Enter your name</label>
  <input type = "text" id="name" name="name"></p>
  <p><label for="age">Enter age</label>
  <input type = "text" id = "age" name="age"></p>
  <p><label for="email">Enter your email:</label>
  <input type="text" id="email" name="email"></p>
  <p><label for="URL">Enter your website<label>
  <input type = "text" id="URL" name="url"></p>
  <input type="submit">
</form>
```

The number type text field shows up/down counter arrows on the right. Only number is accepted, and can be incremented or decremented.

Input Validation

Enter your name

Enter age

Enter your email:

Enter your website

If the data in email field is invalid, you get the error message flashed as below.

Input Validation

Enter your name

Enter age

Enter your email:

 Please include an '@' in the email address. 'asd#xyz.com' is missing an '@'.

Similarly, any incorrect format for the URL also flashes the error as shown –

Input Validation

Enter your name

Enter age

Enter your email:

Enter your website

 Please enter a URL.

Validation Functions

The validation on the server-side with PHP comes into picture, either when the form data passes the client-side validation, or there's no validation on the client side at all.

In the HTML form used in the above example, let us remove all special input types and use all text fields of text type. The form is submitted with POST method to hello.php on the server.

</>

Open Compiler

```
<form action="hello.php" method="POST">
  <p><label for="name">Enter your name</label>
  <input type="text" id="name" name="name"></p>
  <p><label for="age">Enter age</label>
  <input type="text" id="age" name="age"></p>
  <p><label for="email">Enter your email:</label>
  <input type="text" id="email" name="email"></p>
  <p><label for="URL">Enter your website</label>
  <input type="text" id="URL" name="url"></p>
  <input type="submit">
</form>
```

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

Form is Empty

If the user (may be inadvertently) clicks the submit button, you can ask PHP to display the form again. You need to check if the `$_POST` array has been initialized with `isset()` function. If not, the `header()` function redirects the control back to the form.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST)) {
        header("Location: hello.html", true, 301);
        exit();
    }
    // form processing if the form is not empty
}
?>
```

Example

You can also check if any of the fields is empty at the time of submitting the form.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    foreach($_POST as $k=>$v) {
        if (empty($v)==true) {
            echo "One or more fields are empty \n";
            echo "<a href = 'hello.html'>Click here to go back </a>";
            exit;
        }
        else
            echo "$k => $v \n";
    }
}
?>
```

Age field is non-numeric

In the HTML form the input field for name is of text type, hence it can accept any characters. However, we want it to be numeric. This can be ensured by `is_numeric()` function

```
<?php
    if (is_numeric($_POST["age"])==false) {
        echo "Age cannot be non-numeric \n";
        echo "<a href = 'hello.html'>Click here to go back</a>";
    }
?>
```

PHP also has `is_string()` function to check if a file contains a string or not. Two other functions, `trim()` and `htmlspecialchars()` are also useful for form validation.

- **trim()** – Removes whitespace from the beginning and end of a string
- **htmlspecialchars()** – Converts special characters to HTML entities to prevent cross-site scripting (XSS) attacks.