PHP Form Validation

In this tutorial you'll learn how to sanitize and validate form data using PHP filters.

Sanitizing and Validating Form Data

As you have seen in the previous tutorial, the process of capturing and displaying the submitted form data is quite simple. In this tutorial you will learn how to implement a simple contact form on your website that allows the user to send their comment and feedback through email. We will use the same PHP `mail()` function to send the emails.

We are also going to implement some basic security feature like sanitization and validation of the user's input so that user can not insert potentially harmful data that compromise the website security or might break the application.

The following is our all-in-one PHP script which does the following things:

- It will ask the users to enter his comments about the website.
- The same script displays the contact form and process the submitted form data.
- The script sanitizes and validates the user inputs. If any required field (marked with `*`) is missing or validation failed due to incorrect inputs the script redisplays the form with an error message for corresponding form field.
- The script remembers which fields the user has already filled in, and prefills those fields when the form redisplayed due to validation error.
- If the data submitted by the user are acceptable and everything goes well it will send an email to the website administrator and display a success message to the user.

Type the following code in "contact.php" file and save in your project root directory:

Example

Download

```php
<?php
// Functions to filter user inputs
function filterName($field){
    // Sanitize user name
    $field = filter_var(trim($field), FILTER_SANITIZE_STRING);

    // Validate user name
    if(filter_var($field, FILTER_VALIDATE_REGEXP, array("options"=>array("regexp"=>"/^[a-zA-Z\s]+$/")))){
        return $field;
    } else{
        return FALSE;
    }
}
```

Explanation of code

You might think what that code was all about. OK, let's get straight into it.

- The `filterName()` function (*line no-03*) validate input value as person's name. A valid name can only contain alphabetical characters (a-z, A-Z).
- The `filterEmail()` function (*line no-14*) validate input value as email address.
- The `filterString()` function (*line no-25*) only sanitize the input value by stripping HTML tags and special characters. It doesn't validate the input valueagainstanything.
- The attribute `action="contact.php"` (*line no-111*) inside the `<form>` tag specifies that the same `contact.php` file display the form as well as process the form data.
- The PHP code inside the value attribute of `<input>` and `<textarea>` e.g. `<?php echo $name; ?>` display prefilled value when form is redisplayed upon validation error.
- The PHP code inside the `.error` class e.g. `<span class="error"><?php echo $nameErr; ?></span>` display error for corresponding field.

Rest the thing we have already covered in previous chapters. To learn more about sanitize and validate filters, please check out the PHP Filter reference.

Note: You need to setup a mail server on your machine for the PHP `mail()` function to work. If you just want to implement the form validation you can replace the mail part (line no. 81 to 94) with your own custom code.