# PHP Form Handling

In this tutorial you'll learn how to collect user inputs submitted through a form using the PHP superglobal variables `$_GET`, `$_POST` and `$_REQUEST`.

## Creating a Simple Contact Form

In this tutorial we are going to create a simple HMTL contact form that allows users to enter their comment and feedback then displays it to the browser using PHP.

Open up your favorite code editor and create a new PHP file. Now type the following code and save this file as "contact-form.php" in the root directory of your project.

Example

Download

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Contact Form</title>
</head>
<body>
    <h2>Contact Us</h2>
    <p>Please fill in this form and send us.</p>
    <form action="process-form.php" method="post">
        <p>
            <label for="inputName">Name:<sup>*</sup></label>
            <input type="text" name="name" id="inputName">
        </p>
```

## Explanation of code

Notice that there are two attributes within the opening `<form>` tag:

- The `action` attribute references a PHP file "process-form.php" that receives the data entered into the form when user submit it by pressing the submit button.
- The `method` attribute tells the browser to send the form data through POST method.

Rest of the elements inside the form are basic form controls to receive user inputs. To learn more about HTML form elements please check out the HTML Forms tutorial.

# Capturing Form Data with PHP

To access the value of a particular form field, you can use the following superglobal variables. These variables are available in all scopes throughout a script.

| Superglobal | Description |
|---|---|
| `$_GET` | Contains a list of all the field names and values sent by a form using the get method (i.e. via the URL parameters). |
| `$_POST` | Contains a list of all the field names and values sent by a form using the post method (data will not visible in the URL). |
| `$_REQUEST` | Contains the values of both the `$_GET` and `$_POST` variables as well as the values of the `$_COOKIE` superglobal variable. |

When a user submit the above contact form through clicking the submit button, the form data is sent to the "process-form.php" file on the server for processing. It simply captures the information submitted by the user and displays it to browser.

The PHP code of "process-form.php" file will look something like this:

Example

Download

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Contact Form</title>
</head>
<body>
    <h1>Thank You</h1>
    <p>Here is the information you have submitted:</p>
    <ol>
        <li><em>Name:</em> <?php echo $_POST["name"]?></li>
        <li><em>Email:</em> <?php echo $_POST["email"]?></li>
        <li><em>Subject:</em> <?php echo $_POST["subject"]?></li>
        <li><em>Message:</em> <?php echo $_POST["message"]?></li>
    </ol>
</body>
</html>
```

The PHP code above is quite simple. Since the form data is sent through the post method, you can retrieve the value of a particular form field by passing its name to the `$_POST` superglobal array, and displays each field value using `echo()` statement.

In real world you cannot trust the user inputs; you must implement some sort of validation to filter the user inputs before using them. In the next chapter you will learn how sanitize and validate this contact form data and send it through the email using PHP.