

PHP Form Handling

PHP Super Global Variables

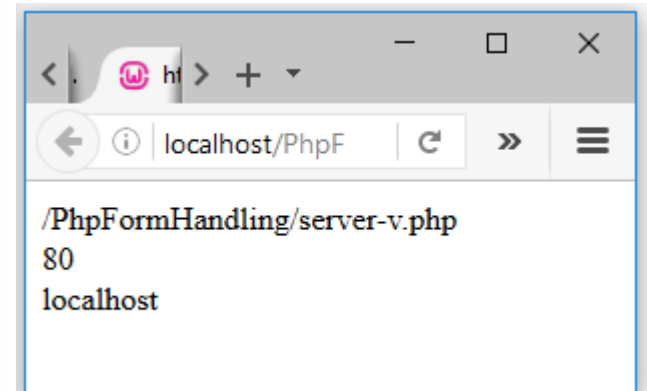
- Super global variables are predefined variables in PHP, which means that they are:
 - always accessible, regardless of scope
 - accessible from any function, class or file.
 - `$_SERVER`, `$_REQUEST`, `$_POST`, and `$_GET` are global variables.

	Variable	Definition
1	GLOBALS	References all variables available in global scope
2	\$_SERVER	Server and execution environment information
3	\$_GET	HTTP GET variables
4	\$_POST	HTTP POST variables
5	\$_FILES	HTTP File upload variables
6	\$_COOKIE	HTTP Cookies
7	\$_SESSION	Session variables
8	\$_REQUEST	HTTP Request variables
9	\$_ENV	Environment variables

Example : PHP \$_SERVER

- \$_SERVER is a PHP super global variable which holds information about server, headers, paths, and script locations.

```
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      echo $_SERVER['PHP_SELF'];
      echo "<br>";
      echo $_SERVER['SERVER_PORT'];
      echo "<br>";
      echo $_SERVER['SERVER_NAME'];
    ?>
  </body>
</html>
```

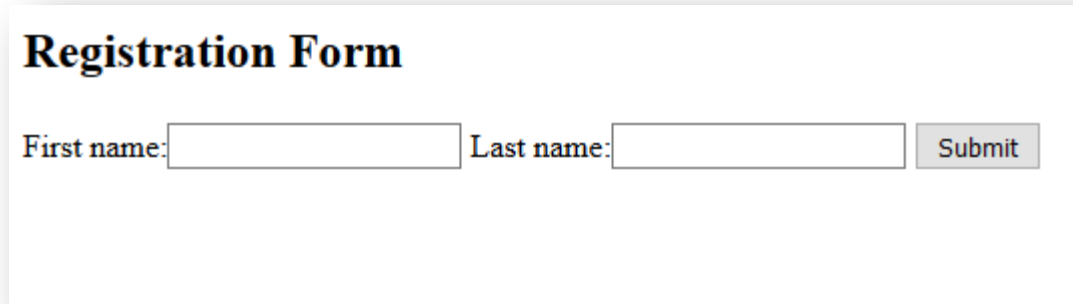


What is a form?

- Forms are used to get input from the user and submit it to the web server for processing.
- A form is an HTML tag that contains graphical user interface items such as input box, check boxes radio buttons.
- The form is defined using the `<form>...</form>` tags and **GUI items** are defined using form elements such as input.

Example:

```
<html>
  <head>
    <title>Registration Form</title>
  </head>
  <body>
    <h2>Registration Form</h2>
    <form action="registration_form.php" method="POST">
      First name:<input type="text" name="firstname">
      Last name:<input type="text" name="lastname">
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```



Registration Form

First name: Last name:

Methods of Sending Information to Server

- The action attribute of the form specifies PHP file that processes the data.
- The method attribute specifies the submission method type.
- A web browser communicates with the server typically using one of the two

HTTP methods:

- **GET**
- **POST**

Submission Method

PHP file that receives data

```
<html>
  <head>
    <title>Example of PHP GET method</title>
  </head>
  <body>
    <form method="get" action="process.php">
      <label for="inputName">Name:</label>
      <input type="text" name="name">
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

The GET Method

- In GET method the data is sent as URL parameters that are usually strings of name and value pairs separated by ampersands (&).
- PHP provides the superglobal variable `$_GET` to access all the information sent either through the URL or submitted through an HTML form using the `method="get"`.

```
http://www.example.com/action.php?name=john&age=24
```


The GET Method (cont)

- The GET method is not suitable for passing sensitive information such as the username and password, because these are fully visible in the URL query string as well as potentially stored in the client browser's memory as a visited page.
- Because the GET method assigns data to a server environment variable, the length of the URL is limited. So, there is a limitation for the total data to be sent.

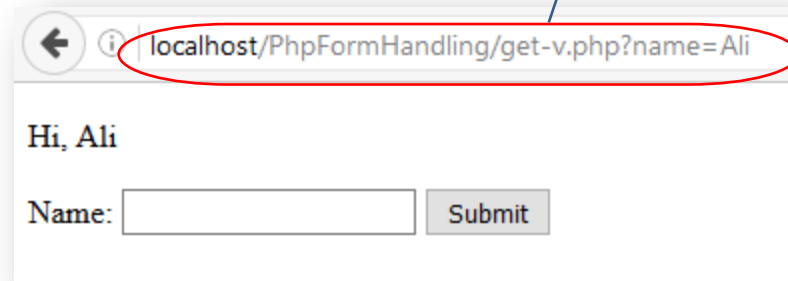
Example (get):

Submission
Method
(get)

PHP file that
receives data

```
<html>
  <head>
    <title>Example of PHP GET method</title>
  </head>
  <body>
    <form method="get" action="process.php">
      <label for="inputName">Name:</label>
      <input type="text" name="name">
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

Form values
visible



The POST Method

- In POST method the data is sent to the server as a package in a separate communication with the processing script.
- Data sent through POST method will not be visible in the URL.
- There is a much larger limit on the amount of data that can be passed and one can send text data as well as binary data using POST.
- PHP provides the superglobal variable `$_POST` to access all the information sent via post method or submitted through an HTML form using the `method="post"`.

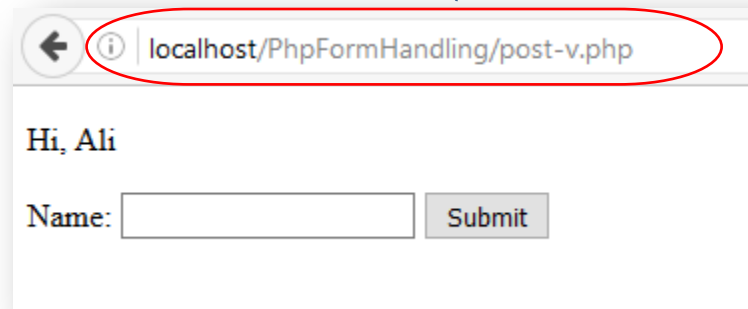
Example (post):

Submission
Method
(post)

PHP file that
receives data

```
<html>
  <head>
    <title>Example of PHP GET method</title>
  </head>
  <body>
    <form method="post" action="process.php">
      <label for="inputName">Name:</label>
      <input type="text" name="name" >
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

Form values
Not visible



Capturing Form Data with PHP

- To access the value of a particular form field, we can use one of the following superglobal variables:
 - `$_GET`
 - `$_POST`
 - `$_REQUEST`

Using Post Method

```
<html>
  <head>
    <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>
        Name:<input type="text" name="name" >
      <p>
        Email <input type="text" name="email" >
      </p>
      <p>
        Subject: <input type="text" name="subject" >
      </p>
      <p>
        Message: <textarea name="message" rows="5" cols="30"></textarea>
      </p>
      <input type="submit" value="Submit">
      <input type="reset" value="Reset">
    </form>
  </body>
</html>
```

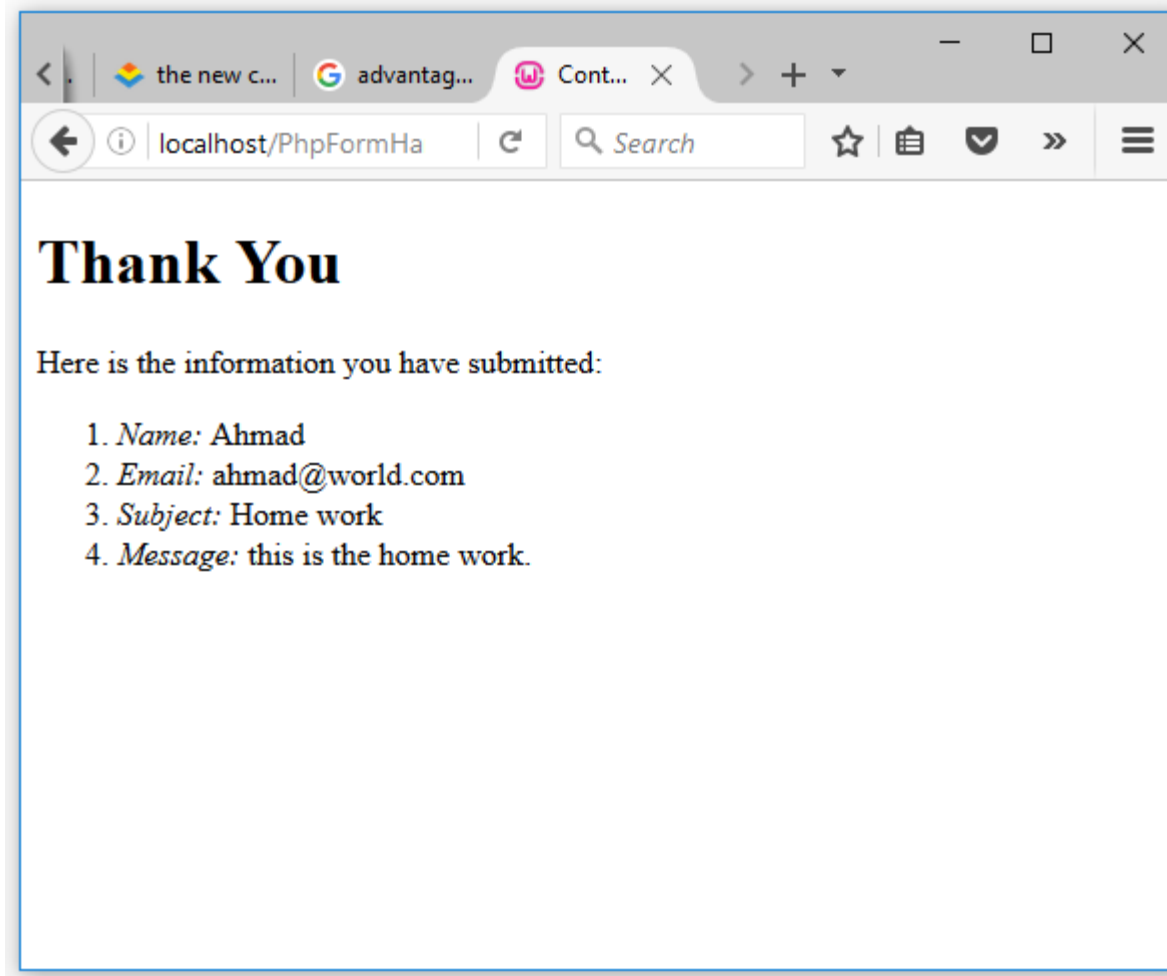

process-form.php

```
<html>
  <head>
    <title>Contact Form</title>
  </head>
  <body>
    <h1>Thank You</h1>
    <p>Here is the information you have submitted:</p>
    <ol>
      <li>Name: <?php echo $_POST["name"] ?></li>
      <li>Email:<?php echo $_POST["email"] ?></li>
      <li>Subject: <?php echo $_POST["subject"] ?></li>
      <li>Message:<?php echo $_POST["message"] ?></li>
    </ol>
  </body>
</html>
```

Using \$_POST

A screenshot of a web browser window displaying a contact form. The browser's address bar shows the URL `localhost/PhpFormHa`. The page title is "Contact Us". The form contains the following elements:

- A heading: **Contact Us**
- A prompt: "Please fill in this form and send us."
- A "Name:" label followed by a text input field containing "Ahmad".
- An "Email" label followed by a text input field containing "ahmad@world.com".
- A "Subject:" label followed by a text input field containing "Home work".
- A "Message:" label followed by a text area containing "this is the home work.".
- Two buttons at the bottom: "Submit" and "Reset".



Using Get Method

```
<html>
  <head>
    <title>Contact Form</title>
  </head>
  <body>
    <h2>Contact Us</h2>
    <p>Please fill in this form and send us.</p>
    <form action="process-form-get.php" method="get">
      <p>
        Name: <input type="text" name="name" >
      </p>
      <p>
        Email <input type="text" name="email" >
      </p>
      <p>
        Subject: <input type="text" name="subject" >
      </p>
      <p>
        Message: <textarea name="message" rows="5" cols="30"></textarea>
      </p>
      <input type="submit" value="Submit">
      <input type="reset" value="Reset">
    </form>
  </body>
</html>
```

process-form-get.php

```
<html>
  <head>
    <title>Contact Form</title>
  </head>
  <body>
    <h1>Thank You</h1>
    <p>Here is the information you have submitted:</p>
    <ol>
      <li>Name:<?php echo $_GET["name"] ?></li>
      <li>Email:<?php echo $_GET["email"] ?></li>
      <li>Subject: <?php echo $_GET["subject"] ?></li>
      <li>Message: <?php echo $_GET["message"] ?></li>
    </ol>
  </body>
</html>
```

Using \$_GET

The \$_REQUEST Variable

- PHP \$_REQUEST is used to collect data after submitting an HTML form.
- The \$_REQUEST contains the values of both the \$_GET and \$_POST variables as well as the values of the \$_COOKIE variable.

```
<html>
  <head>
    <title>Example of PHP $_REQUEST variable</title>
  </head>
  <body>
    <form method="post" action="process.php">
      <label for="inputName">Name:</label>
      <input type="text" name="name" id="inputName">
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

```
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
    echo "<p>Hi, " . $_REQUEST["name"] . "</p>";
    ?>
  </body>
</html>
```

Thanks!