PHP:  
PHP stands for Hypertext Preprocessor. PHP is a very popular and widely-used open source server-side scripting language to write dynamically generated web pages. PHP was originally created by Rasmus Lerdorf in 1994. It was initially known as Personal Home Page.

PHP scripts are executed on the server and the result is sent to the web browser as plain HTML. PHP can be integrated with the number of popular databases, including MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Sybase, and so on. The current major version of PHP is 7. All of the code in this tutorial has been tested and validated against the most recent release of PHP 7.

PHP is very powerful language yet easy to learn and use. So bookmark this website and continued on.

**What You Can Do with PHP**

There are lot more things you can do with PHP.

* You can generate pages and files dynamically.
* You can create, open, read, write and close files on the server.
* You can collect data from a web form such as user information, email, phone no, etc.
* You can send emails to the users of your website.
* You can send and receive cookies to track the visitor of your website.
* You can store, delete, and modify information in your database.
* You can restrict unauthorized access to your website.
* You can encrypt data for safe transmission over internet.

**Advantages of PHP over Other Languages**

If you're familiar with other server-side languages like ASP.NET or Java, you might be wondering what makes PHP so special. There are several advantages why one should choose PHP.

* **Easy to learn:** PHP is easy to learn and use. For beginner programmers who just started out in web development, PHP is often considered as the preferable choice of language to learn.
* **Open source:** PHP is an open-source project. It is developed and maintained by a worldwide community of developers who make its source code freely available to download and use.
* **Portability:** PHP runs on various platforms such as Microsoft Windows, Linux, Mac OS, etc. and it is compatible with almost all servers used today such Apache, IIS, etc.
* **Fast Performance:** Scripts written in PHP usually execute or runs faster than those written in other scripting languages like ASP, Ruby, Python, Java, etc.
* **Vast Community:** Since PHP is supported by the worldwide community, finding help or documentation related to PHP online is extremely easy.

**Setting Up a Local Web Server**

PHP script execute on a web server running PHP. So before you start writing any PHP program you need the following program installed on your computer.

* The Apache Web server
* The PHP engine
* The MySQL database server

## Creating Your First PHP Script

<?php // Display greeting message

echo "Hello, world!";

?>

PHP in HTML:

PHP can be embedded within a normal HTML web page. That means inside your HTML document you can write the PHP statements, as demonstrated in the follwoing example:

#### Example

[**Run this code »**](https://www.tutorialrepublic.com/codelab.php?topic=php&file=simple-document)

<!DOCTYPE HTML>

<html>

<head>

<title>PHP Application</title>

</head>

<body>

<?php

// Display greeting message

echo 'Hello World!';

?>

</body>

</html>

## What is Variable in PHP

Variables are used to store data, like string of text, numbers, etc. Variable values can change over the course of a script. Here're some important things to know about variables:

* In PHP, a variable does not need to be declared before adding a value to it. PHP automatically converts the variable to the correct data type, depending on its value.
* After declaring a variable it can be reused throughout the code.
* The assignment operator (=) used to assign value to a variable.

In PHP variable can be declared as: $var\_name = value;

#### Example

[**Run this code »**](https://www.tutorialrepublic.com/codelab.php?topic=php&file=variables)

1.<?php

// Declaring variables

$txt = "Hello World!";

$number = 10;

// Displaying variables value

echo $txt; // Output: Hello World!

echo $number; // Output: 10

?>

2.<?php

$x = 10; $y = 4;

echo($x + $y); // 0utputs: 14

echo($x - $y); // 0utputs: 6

echo($x \* $y); // 0utputs: 40

echo($x / $y); // 0utputs: 2.5

echo($x % $y); // 0utputs: 2

?>

**Naming Conventions for PHP Variables**

These are the following rules for naming a PHP variable:

* All variables in PHP start with a $ sign, followed by the name of the variable.
* A variable name must start with a letter or the underscore character \_.
* A variable name cannot start with a number.
* A variable name in PHP can only contain alpha-numeric characters and    underscores (A-z, 0-9, and \_).
* A variable name cannot contain spaces.

**Types of Arrays in PHP**

There are three types of arrays that you can create. These are:

* **Indexed array** — An array with a numeric key.
* **Associative array** — An array where each key has its own specific value.
* **Multidimensional array** — An array containing one or more arrays within itself.

## Indexed Arrays

An indexed or numeric array stores each array element with a numeric index. The following examples shows two ways of creating an indexed array, the easiest way is:

#### Example

[**Run this code »**](https://www.tutorialrepublic.com/codelab.php?topic=php&file=indexed-array-01)

<?php

// Define an indexed array

$colors = array("Red", "Green", "Blue");

?>

## Associative Arrays

In an associative array, the keys assigned to values can be arbitrary and user defined strings. In the following example the array uses keys instead of index numbers:

<?php

$variable\_name['key\_name'] = value;

$variable\_name = array('keyname' => value);

?>

HERE,

* “$variable\_name…” is the name of the variable
* “['key\_name']” is the access index number of the element
* “value” is the value assigned to the array element.

#### Example

[**Run this code »**](https://www.tutorialrepublic.com/codelab.php?topic=php&file=associative-array-01)

<?php

// Define an associative array

$ages = array("Peter"=>22, "Clark"=>32, "John"=>28);

?>

The following example is equivalent to the previous example, but shows a different way of creating associative arrays:

#### Example

[**Run this code »**](https://www.tutorialrepublic.com/codelab.php?topic=php&file=associative-array-02)

<?php

$ages["Peter"] = "22";

$ages["Clark"] = "32";

$ages["John"] = "28";

?>

## Multidimensional Arrays

The multidimensional array is an array in which each element can also be an array and each element in the sub-array can be an array or further contain array within itself and so on. An example of a multidimensional array will look something like this:

#### Example

[**Run this code »**](https://www.tutorialrepublic.com/codelab.php?topic=php&file=multidimensional-array)

<?php

// Define a multidimensional array

$contacts = array(

array(

"name" => "Peter Parker",

"email" => "peterparker@mail.com",

),

array(

"name" => "Clark Kent",

"email" => "clarkkent@mail.com",

),

array(

"name" => "Harry Potter",

"email" => "harrypotter@mail.com",

)

);

// Access nested value

echo "Peter Parker's Email-id is: " . $contacts[0]["email"];

?>

## PHP foreach Loop

The foreach loop is used to iterate over arrays.

foreach($array as $value){  
    // Code to be executed  
}

The following example demonstrates a loop that will print the values of the given array:

#### Example

[**Run this code »**](https://www.tutorialrepublic.com/codelab.php?topic=php&file=foreach-loop)

<?php

// Declare an array

$arr = array("green", "blue", "pink", "white");

// Loop through the array elements

foreach ($arr as $element) {

    echo "$element ";

}

?>

**Output:**

green blue pink white

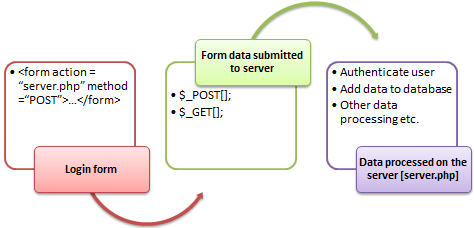
# PHP Registration Form using GET, POST Methods with Example

## What is Form?

When you login into a website or into your mail box, you are interacting with a form.

Forms are used to get input from the user and submit it to the web server for processing.

 The diagram below illustrates the form handling process.

[](https://www.guru99.com/images/2013/04/php_forms.png)

A form is an HTML tag that contains graphical user interface items such as input box, check boxes radio buttons etc.

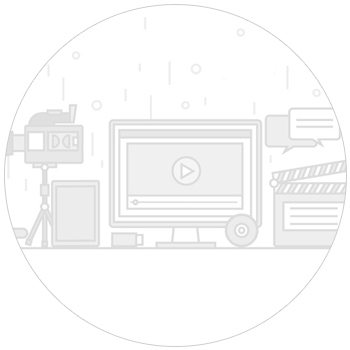
The form is defined using the <form>...</form> tags and GUI items are defined using form elements such as input.

## When and why we are using forms?

* Forms come in handy when developing flexible and dynamic applications that accept user input.
* Forms can be used to edit already existing data from the database

## Create a form

We will use HTML tags to create a form. Below is the minimal list of things you need to create a form.



Seven Testing Principles Software Testing

* Opening and closing form tags <form>…</form>
* Form submission type POST or GET
* Submission URL that will process the submitted data
* Input fields such as input boxes, text areas, buttons,checkboxes etc.

**The code below creates a simple registration form**

<html>

<head>

<title>Registration Form</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

<h2>Registration Form</h2>

<form action="registration\_form.php" method="POST"> First name:

<input type="text" name="firstname"> <br> Last name:

<input type="text" name="lastname">

<input type="hidden" name="form\_submitted" value="1" />

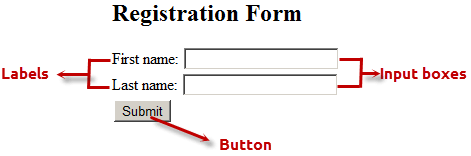
<input type="submit" value="Submit">

</form>

</body>

</html>

Viewing the above code in a web browser displays the following form.

[](https://www.guru99.com/images/2013/04/html_form.png)

HERE,

* <form…>…</form> are the opening and closing form tags
* action="registration\_form.php" method="POST"> specifies the destination URL and the submission type.
* First/Last name: are labels for the input boxes
* <input type=”text”…> are input box tags
* <br> is the new line tag
* <input type="hidden" name="form\_submitted" value="1"/> is a hidden value that is used to check whether the form has been submitted or not
* <input type="submit" value="Submit"> is the button that when clicked submits the form to the server for processing

## Submitting the form data to the server

The action attribute of the form specifies the submission URL that processes the data. The method attribute specifies the submission type.

### PHP POST method

* This is the built in PHP super global array variable that is used to get values submitted via HTTP POST method.
* The array variable can be accessed from any script in the program; it has a global scope.
* This method is ideal when you do not want to display the form post values in the URL.
* A good example of using post method is when submitting login details to the server.

**It has the following syntax.**

<?php

$\_POST['variable\_name'];

?>

  HERE,

* “$\_POST[…]” is the PHP array
* “'variable\_name'” is the URL variable name.

### PHP GET method

* This is the built in PHP super global array variable that is used to get values submitted via HTTP GET method.
* The array variable can be accessed from any script in the program; it has a global scope.
* This method displays the form values in the URL.
* It’s ideal for search engine forms as it allows the users to book mark the results.

**It has the following syntax.**

<?php

$\_GET['variable\_name'];

?>

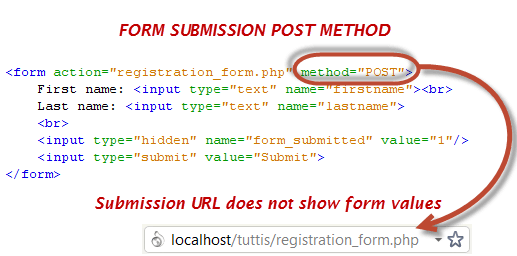
  HERE,

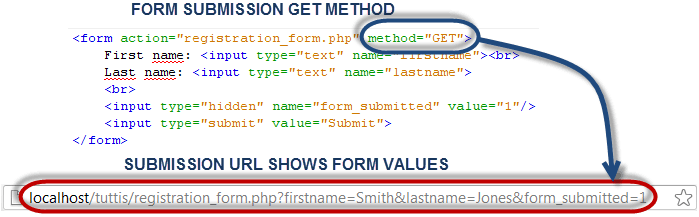
* “$\_GET[…]” is the PHP array
* “'variable\_name'” is the URL variable name.

## GET vs POST Methods

|  |  |
| --- | --- |
| **POST** | **GET** |
| Values not visible in the URL | Values visible in the URL |
| Has not limitation of the length of the values since they are submitted via the body of HTTP | Has limitation on the length of the values usually 255 characters. This is because the values are displayed in the URL. Note the upper limit of the characters is dependent on the browser. |
| Has lower performance compared to Php\_GET method due to time spent encapsulation the Php\_POST values in the HTTP body | Has high performance compared to POST method dues to the simple nature of appending the values in the URL. |
| Supports many different data types such as string, numeric, binary etc. | Supports only string data types because the values are displayed in the URL |
| Results cannot be book marked | Results can be book marked due to the visibility of the values in the URL |

**The below diagram shows the difference between get and post**

[](https://www.guru99.com/images/2013/04/post_form_submission.png)

[](https://www.guru99.com/images/2013/04/get_form_submission.png)