

Share on your Social Media



PHP Tutorial

Published On: July 1, 2024

PHP Tutorial

PHP, or Hypertext Preprocessor, is a widely used server-side scripting language for creating interactive web pages. This PHP tutorial is designed for beginners to provide a complete understanding of developing dynamic web pages efficiently using PHP.

[Download PHP tutorial PDF](#)

Introduction to PHP

The expression "PHP means hypertext preprocessor." One popular open-source programming language is PHP. PHP scripts run on the server. You may download and use PHP for free. PHP is useful in the following ways:

- PHP is capable of producing dynamic webpage content.
- PHP may create, open, read, write, delete, and close files on the server.
- PHP can gather form data.
- PHP can transmit and receive cookies.
- PHP may change, add, and remove data from your database.
- PHP is a tool for managing user access.
- PHP has data encryption capabilities.
- PHP allows you to output more than just HTML.



Featured Articles

Want to know more about becoming an expert in IT?

Click Here to Get Started >>

100% Placement Assurance

AUTHORISED CERTIFICATION PARTNER **IBI**



Related Courses at SLA

- ➔ **PHP Training in Chennai**
- ➔ **PHP Online Training**

Related Posts

C and C++ Tutorial
Published On: August 1, 2024

C and C++ Tutorial C is a high-level, procedural,

PDF files or photos can be produced. Additionally, you can export any text, including XHTML and XML.

- PHP 7 is significantly faster than PHP 5.6, the last stable and widely used version.
- PHP 7 has made error handling better.
- More stringent type declarations for function parameters are supported by PHP 7.
- PHP 7 supports new operators (like the spaceship operator: `<=>`).

PHP Installation

To begin with PHP, you may:

- Look for a web host that supports MySQL and PHP.
- Install PHP and MySQL after setting up a web server on your personal computer.

Three essential components must be installed on your computer system to create and execute PHP Web pages.

Web Server: PHP is compatible with almost every web server software, such as Lighttpd, NGINX, and Microsoft's Internet Information Server (IIS).

Get Apache for free by visiting this link:
<https://httpd.apache.org/download.cgi>

Database: PHP is compatible with almost any database program, such as Sybase and Oracle, although MySQL is the most often used free database.

Get MySQL for free by visiting this link:
<https://www.mysql.com/downloads/>

PHP Parser: To process PHP script instructions and produce HTML output that can be seen in a web browser, a parser needs to be installed.

Installing XAMP

XAMPP is an Apache distribution that is simple to

general-purpose programming language. Whereas C++, a...



ASP DOTNET Tutorial

Published On: July 31, 2024

ASP DOTNET Tutorial Microsoft created the web framework known as ASP.NET. It is employed in...



Artificial Intelligence Tutorial

Published On: July 30, 2024

Artificial Intelligence Tutorial Artificial intelligence (AI) is significant since it enhances many facets of society...



Appium Testing Tutorial

Published On: July 30, 2024

Appium Testing Tutorial Designed to make the UI automation of many app platforms easier, Appium...

install and comes with Apache, MariaDB, PHP, and Perl. The acronym's letter X denotes that the program is cross-platform, working with Linux, OS X, and Windows.

It should be noted that MariaDB, an identically functioning clone of MySQL, is included with XAMPP.

Visit <https://www.apachefriends.org/download.html> to obtain the appropriate installer for your operating system.

Download one of the following:

Windows - <https://sourceforge.net/projects/>

Linux - <https://sourceforge.net/projects/>

OS X - <https://sourceforge.net/projects/>

[Download PHP Syllabus PDF](#)

Features of PHP

New features and code changes are often added to PHP versions. The following are PHP's salient features:

- Simple and easy to learn.
- Open-source to use, modify, and distribute.
- Cross-platform compatible
- Server-side scripting
- Easy integration with various databases
- Extensive library support
- Built-in security features
- Efficient memory and session management in PHP.
- Active community support

PHP Syntax

PHP syntax is quite similar to C language syntax. A text file with the ending ".php" contains a PHP code. A ".php" file is a web page containing one or more PHP code blocks scattered throughout the HTML

script.

PHP File: PHP code, HTML, CSS, JavaScript, and text can all be found in PHP files. After PHP code is run on the server, plain HTML is sent back to the browser. “.php” is the extension for PHP files.

Canonical PHP Tags

The PHP tag style that works best everywhere is:

```
<?php
```

One or more PHP statements

```
?>
```

SGML-Style (Short-open) Tags

The shortest option is, naturally, a short tag. It looks as follows:

```
<?php
```

One or more PHP statements

```
?>
```

To allow PHP to identify the tags, you need to do one of two things:

- When generating PHP, select the “–enable-short-tags” configuration option.
- In your php.ini file, set the “short_open_tag” value to on.

```
short_open_tag=on
```

Since XML tags utilize the same syntax, this option needs to be disabled to parse XML with PHP. The implementation of ASP-style tags is as follows:

```
<%...%>
```

and script tags for HTML `<script language = “PHP”>...</script>` is no longer in use.

Basic Syntax

PHP shares a lot of basic syntax similarities with C and C++.

- In PHP, an expression is considered a statement if it ends with **a semicolon (;)**.

```
$greeting = "Welcome to PHP!";
```

- **Token combinations are used in PHP expressions:** PHP's indivisible tokens, which include numbers (3.14159), strings ("two"), variables (\$two), constants (TRUE), and special words like "if", "else", "while", "for", and so on, are its tiniest building blocks.
- **Braces create building blocks:** You can always put a series of statements anywhere a statement can go by surrounding them in a set of curly braces, even if statements cannot be joined like expressions.

```
if (3 == 2 + 1)  
    print("SLA – The best place for IT training");  
  
if (3 == 2 + 1) {  
    print("SLA – The best place");  
    print("for IT training");  
}
```

- **PHP is case-sensitive:** Case matters when naming different PHP identifiers, such as variables, functions, classes, etc.

Consequently, "\$age" and "\$Age" are not the same variables. In a similar vein, "myfunction()" and "MyFunction()" are two separate functions.

Comments in PHP

A line that is not executed as part of the program is called a comment in PHP code. Its sole function is to allow whoever is viewing the code to read it.

You can use comments for:

- Make your code understandable to others.

- Remind yourself of your actions: The majority of programmers have had to relearn how to do tasks after returning to their own employment after a year or two. Comments serve as a reminder of your thoughts at the time the code was written.
- Remove a few lines from your code.

PHP allows for multiple comment formats:

Example

```
// This is a single-line comment
```

```
# This is also a single-line comment
```

```
/* This is a
```

```
multi-line comment */
```

PHP Variables

Information is stored in “containers” called variables.

Creating PHP Variables

Variables in PHP begin with the \$ symbol and are named after that:

```
$x = 1;
```

```
$y = "Raj"
```

The number 1 will be stored in the variable \$x in the example above, and the value “Raj” will be stored in the variable \$y.

Note: Enclose the text value in quotes when you assign it to a variable.

PHP lacks a command for declaring variables, in contrast to most programming languages. The instant you give it a value, it is produced.

A variable’s name can be short (\$x, \$y) or long (\$age, \$carname, \$total_volume) to provide more information about it.

Rules for PHP variables:

- A variable's name comes after the \$ symbol at the beginning.
- An underscore character or a letter must appear at the beginning of a variable name.
- You cannot begin a variable name with a number.
- Only alpha-numeric characters and underscores (A-z, 0-9, and _) are permitted in a variable name.
- Case matters when naming variables (\$age and \$AGE are two distinct variables).

Example

```
$x = 5;
```

```
$y = 4;
```

```
echo $x + $y;
```

PHP Data Types

Different forms of data can be stored in variables, and different types of data have distinct functions.

PHP supports the following data types:

- String
- Integer
- Float (floating point numbers – also called double)
- Boolean
- Array
- Object
- NULL
- Resource

The `var_dump()` function allows you to obtain the data type of any object.

Example

```
$x = 5;
```

```
var_dump($x);
```

String in PHP

A string, such as "Hello world!" is a series of characters. Anything within quotations can be a string. You may utilize a single quote or two:

```
$x = "Hello world!";
```

```
$y = 'Hello world!';
```

```
var_dump($x);
```

```
echo "<br>";
```

```
var_dump($y);
```

Integer in PHP

Any non-decimal number between -2,147,483,648 and 2,147,483,647 is an integer data type.

Guidelines for integers:

- There must be one digit in an integer.
- There cannot be a decimal point in an integer.
- A number can have a positive or negative value.
- Decimal (base 10), hexadecimal (base 16), octal (base 8), and binary (base 2) are the possible bases in which to specify integers.

Notation

\$x\$ is an integer in the example that follows. The PHP `var_dump()` method returns the data type and value:

```
$x = 5985;
```

```
var_dump($x);
```

Float in PHP

A number in exponential form or one with a decimal point, is called a float (floating point number).

\$x is a float in the example that follows. The PHP

`var_dump()` method returns the data type and value:

```
$x = 10.365;
```

```
var_dump($x);
```

Boolean in PHP

True and false are the two potential states that a boolean represents.

```
$x = true;
```

```
var_dump($x);
```

Array in PHP

A single variable can hold several values in an array.

In the example that follows, `$cars` is an array. The PHP `var_dump()` method returns the data type and value:

```
$cars = array("Volvo","BMW","Toyota");
```

```
var_dump($cars);
```

PHP Object

The two primary components of object-oriented programming are classes and objects.

An object is an instance of a class, and a class is a template for objects.

All of the class's behaviors and properties are inherited by the individual objects upon creation; however, the values of each property will vary.

Example

```
class Car {  
  
    public $color;  
  
    public $model;
```

```
public function __construct($color, $model) {  
  
    $this->color = $color;  
  
    $this->model = $model;  
  
}  
  
public function message() {  
  
    return "My car is a ". $this->color . " ". $this->model . "!";  
  
}  
  
}  
  
$myCar = new Car("red", "Volvo");  
  
var_dump($myCar);
```

Null Value in PHP

A unique data type called null has exactly one possible value: NULL. A variable with no value allocated to it is of the data type NULL.

Example

```
$x = "Hello world!";  
  
$x = null;  
  
var_dump($x);
```

PHP Interview Questions and Answers

PHP Operators

Values and variables can be operated on using operators. The operators in PHP are divided into the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators

- Logical operators
- String operators
- Array operators
- Conditional assignment operators

Arithmetic Operators in PHP

Common arithmetical operations like addition, subtraction, multiplication, and so on are carried out using the PHP arithmetic operators with numeric values.

| Operator | Name | Example | Result |
|----------|----------------|--------------|--|
| + | Addition | $\$x + \y | Sum of $\$x$ and $\$y$ |
| - | Subtraction | $\$x - \y | Difference of $\$x$ and $\$y$ |
| * | Multiplication | $\$x * \y | Product of $\$x$ and $\$y$ |
| / | Division | $\$x / \y | Quotient of $\$x$ and $\$y$ |
| % | Modulus | $\$x \% \y | Remainder of $\$x$ divided by $\$y$ |
| ** | Exponentiation | $\$x ** \y | Result of raising $\$x$ to the $\$y$ 'th power |

PHP tutorial 1

Assignment Operators in PHP

To assign a value to a variable that contains numeric values, utilize the PHP assignment operators.

"=" is the fundamental assignment operator in PHP. It indicates that the value of the assignment expression on the right is assigned to the operand on the left.

| Assignment | Same as... | Description |
|---------------------|------------------------|---|
| <code>x = y</code> | <code>x = y</code> | The left operand gets set to the value of the expression on the right |
| <code>x += y</code> | <code>x = x + y</code> | Addition |
| <code>x -= y</code> | <code>x = x - y</code> | Subtraction |
| <code>x *= y</code> | <code>x = x * y</code> | Multiplication |
| <code>x /= y</code> | <code>x = x / y</code> | Division |
| <code>x %= y</code> | <code>x = x % y</code> | Modulus |

PHP Tutorial 2

Comparison Operators in PHP

To compare two values (strings or numbers), use the comparison operators in PHP:

| Operator | Name | Example | Result |
|------------------------|--------------------------|--------------------------------|--|
| <code>==</code> | Equal | <code>\$x == \$y</code> | Returns true if \$x is equal to \$y |
| <code>===</code> | Identical | <code>\$x === \$y</code> | Returns true if \$x is equal to \$y, and they are of the same type |
| <code>!=</code> | Not equal | <code>\$x != \$y</code> | Returns true if \$x is not equal to \$y |
| <code><></code> | Not equal | <code>\$x <> \$y</code> | Returns true if \$x is not equal to \$y |
| <code>!==</code> | Not identical | <code>\$x !== \$y</code> | Returns true if \$x is not equal to \$y, or they are not of the same type |
| <code>></code> | Greater than | <code>\$x > \$y</code> | Returns true if \$x is greater than \$y |
| <code><</code> | Less than | <code>\$x < \$y</code> | Returns true if \$x is less than \$y |
| <code>>=</code> | Greater than or equal to | <code>\$x >= \$y</code> | Returns true if \$x is greater than or equal to \$y |
| <code><=</code> | Less than or equal to | <code>\$x <= \$y</code> | Returns true if \$x is less than or equal to \$y |
| <code><=></code> | Spaceship | <code>\$x <=> \$y</code> | Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. |

PHP Tutorial 3

Increment or Decrement Operators in PHP

The value of a variable can be increased by using the PHP increment operators. The value of a variable can be decreased by using the PHP decrement operators.

PHP Tutorial 4

Logical Operators in PHP

Conditional statements are combined using the PHP logical operators.

PHP Tutorial 5

String Operators in PHP

Two operators in PHP are specifically made for strings.

PHP Tutorial 6

Array Operators in PHP

To compare arrays, utilize the PHP array operators.

PHP Tutorial 7

Conditional Assignment Operators in PHP

To set a value based on circumstances, utilize the PHP conditional assignment operators:

PHP Tutorial 8

[PHP Training](#)

Control Statements in PHP

Writing code frequently involves wanting to execute multiple operations under various circumstances. To do this, you can use conditional statements in your code.

The following conditional statements are available in PHP:

if statement: runs a piece of code if a particular condition is met.

if...else statement: performs a code execution if a condition is true and a different code execution if the condition is false.

if...elseif...else statement: runs distinct scripts under more than two circumstances.

switch statement: chooses one of the numerous code blocks to be run.

If Statement

If one of the conditions in the if statement is true, some code is executed.

Syntax

```
if (condition) {  
    // code to be executed if condition is true;  
}
```

Example

```
if (5 > 3) {  
    echo "Have a good day!";  
}
```

Output

Have a good day!

If-Else Statement

If a condition is true, the if...else statement runs some code; if it is false, it runs another piece of code.

Syntax

```
if (condition) {  
    // code to be executed if condition is true;  
} else {  
    // code to be executed if condition is false;  
}
```

Example

```
$t = date("H");  
if ($t < "20") {  
    echo "Have a good day!";  
} else {
```



```
    echo "Have a good night!";  
}
```

Output

Have a good day!

The if...elseif...else Statement

When more than two conditions are met, separate programs are executed by the if...elseif...else statement.

Syntax

```
if (condition) {  
    code to be executed if this condition is true;  
}  
elseif (condition) {  
    // code to be executed if first condition is false and this condition is  
    true;  
}  
else {  
    // code to be executed if all conditions are false;  
}
```

Example

```
$t = date("H");  
if ($t < "10") {  
    echo "Have a good morning!";  
}  
elseif ($t < "20") {  
    echo "Have a good day!";  
}  
else {  
    echo "Have a good night!";  
}
```

Output

The hour (of the server) is 10, and will give the following message:

Have a good day!

Switch Statement in PHP

Depending on certain circumstances, the switch statement can be used to execute various actions.

The switch statement can be used to choose which of several code blocks should be run.

Syntax

```
switch (expression) {  
  
    case label1:  
  
        //code block  
  
        break;  
  
    case label2:  
  
        //code block;  
  
        break;  
  
    case label3:  
  
        //code block  
  
        break;  
  
    default:  
  
        //code block  
  
}
```

Example

```
$favcolor = "red";  
  
switch ($favcolor) {  
  
    case "red":  
  
        echo "Your favorite color is red!";  
  
        break;  
  
    case "blue":  
  
        echo "Your favorite color is blue!";  
  
        break;  
  
    case "green":  
  
        echo "Your favorite color is green!";  
  
        break;
```

```
default:
    echo "Your favorite color is neither red, blue, nor green!";
}
```

Output

Your favorite color is red!

PHP Developer Salary

Loops in PHP

The same block of code can be repeatedly run using loops as long as a particular condition is met.

The following loop types exist in PHP:

While: iterates across a block of code if the given condition is met

do...While: executes a loop through a block of code once as long as the given condition holds true

For loop: iterates across a code block a predetermined number of times.

Foreach: in an array, the foreach function iterates through a block of code.

While Loop in PHP

The while loop carries out a block of code repeatedly for as long as the given condition holds. The while loop runs a block of code as long as the specified condition is true.

Example

```
$i = 1;
while ($i < 6) {
    echo $i;
    $i++;
}
```

Output

12345

The break Statement

Even if the condition is still true, we can end the loop by using the break statement:

Example

```
$i = 1;
while ($i < 6) {
    if ($i == 3) break;
    echo $i;
    $i++;
}
```

Output

12

Do-While Loop in PHP

The do...while loop always runs the code block through at least one execution, checks the condition, and then continues the loop as long as the given condition is true.

Example

```
$i = 1;
do {
    echo $i;
    $i++;
} while ($i < 6);
```

Output

For Loop in PHP

The for loop iterates over a code block a predetermined number of times. When you know how many times the script should run, you use the for loop.

Syntax

```
for (expression1, expression2, expression3) {  
    // code block  
}
```

Example

```
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}
```

Output

The number is: 0

The number is: 1

The number is: 2

The number is: 3

The number is: 4

The number is: 5

The number is: 6

The number is: 7

The number is: 8

The number is: 9

The number is: 10

For Each Loop in PHP

For each element in an array or property in an object, the foreach loop loops through a block of code. Looping through the elements of an array is the most popular application of the foreach loop.

Example

```
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $x) {
```

```
echo "$x <br>";  
  
}
```

PHP Functions

In addition to the more than 1000 built-in functions in PHP, you may also write your own custom functions.

Build-in Functions in PHP

More than a thousand built-in PHP functions can be used directly to carry out certain tasks from within scripts.

User-Defined Functions in PHP

You can write custom PHP functions in addition to the built-in ones.

A program's function is a set of statements that can be used again and again.

Not every function will start running as soon as a page is loaded.

A call to a function will cause it to be performed.

Create a Function

The function name is placed after the keyword "function" in a user-defined function declaration:

Example

```
function myMessage() {  
  
    echo "Hello world!";  
  
}
```

Call a Function in PHP

Simply type the function's name and parenthesis() to invoke it:

Example

```
function myMessage() {  
  
    echo "Hello world!";  
  
}  
  
myMessage();
```

Arrays in PHP

An array is a unique type of variable that can store multiple values under a single name. By using an index number or name, you can retrieve the items from an array.

An array allows you to store multiple values in a single variable:

```
$cars = array("Tata", "Suzuki", "Hyundai");
```

Types of Array in PHP

There are three kinds of arrays in PHP:

- Numerical index arrays are known as **indexed arrays**.
- Named key arrays, or **associative arrays**
- Arrays that contain one or more arrays are known as **multidimensional arrays**.

Conclusion

We hope that this PHP tutorial will help you get started with your web development learning. Take advantage of a multitude of options by signing up for our [PHP training in Chennai](#).

Share on your Social
Media



Navigation

About Us

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.

Landmark: Karnataka Bank Building

Phone: [+91 86818 84318](tel:+918681884318)

Email: enquiry@softlogicsys.in

Map: [Google Maps Link](#)

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai – 600 130.

Landmark: Adj. to AGS Cinemas

Phone: [+91 89256 88858](tel:+918925688858)

Email: info@softlogicsys.in

Map: [Google Maps Link](#)

Courses

Python

Software Testing

Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

[Blog Posts](#)

[Careers](#)

[Contact](#)

[Placement Training](#)

[Corporate Training](#)

[Hire With Us](#)

[Job Seekers](#)

[SLA's Recently Placed Students](#)

[Reviews](#)

[Sitemap](#)

Important Links

[Disclaimer](#)

[Privacy Policy](#)

[Terms and Conditions](#)

Social Media Links



Review Sources

[Google](#)

[Trustpilot](#)

[Glassdoor](#)

[Mouthshut](#)

[Sulekha](#)

[Justdial](#)

[Ambitionbox](#)

[Indeed](#)

[Software Suggest](#)

