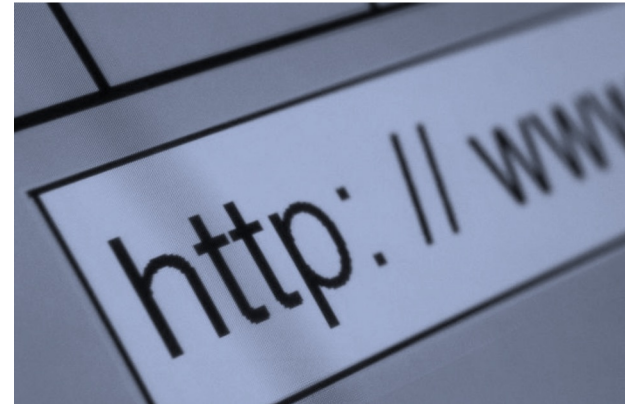


➤ PHP

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.
- PHP 7 is the latest stable release.





To Test your **PhP** code you can install locally on your pc a **PhP** compiler or by using the online compiler on the following link:

https://www.w3schools.com/php/phptryit.asp?filename=tryphp_compiler

PHP Syntax

Basic PHP Syntax

- › A PHP script can be placed anywhere in the document.
- › A PHP script starts with `<?php` and ends with `?>`:

```
<?php
// PHP code goes here
?>
```
- › The default file extension for PHP files is ".php".
- › A PHP file normally contains HTML tags, and some PHP scripting code.

PHP Syntax

Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```



PHP Syntax

PHP Case Sensitivity

In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.

In the example below, all three echo statements below are equal and legal:

```
<?php  
ECHO "Hello World!<br>";  
echo "Hello World!<br>";  
Echo "Hello World!<br>";  
?>
```



PHP Syntax

PHP Case Sensitivity

Note: However; all variable names are case-sensitive!

Look at the example below; only the first statement will display the value of the **\$color** variable! This is because \$color, \$COLOR, and \$coLOR are treated as three different variables:

```
<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>
```

PHP Comments

A comment in PHP code is a line that is not executed as a part of the program.

PHP supports several ways of commenting:

› Syntax for single-line comments:

```
<?php
```

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

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

```
?>
```

PHP Comments

- › Syntax for multiple-line comments:

```
<?php
```

```
/*
```

```
This is a multiple-lines comment block  
that spans over multiple
```

```
lines
```

```
*/
```

```
?>
```

PHP Comments

- Using comments to leave out parts of the code:

```
<?php
```

```
// You can also use comments to leave out parts of a code line
```

```
$x = 5 /* + 15 */ + 5;
```

```
echo $x;
```

```
?>
```

PHP Variables

Variables are "containers" for storing information.

In PHP, a variable starts with the **\$** sign, followed by the name of the variable:

```
<?php  
$txt = "Hello world!";  
$x = 5;  
$y = 10.5;  
?>
```

Note: When you assign a text value to a variable, put quotes around the value.

Note: Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.

PHP Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

Rules for PHP variables:

- › A variable starts with the \$ 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 can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- › Variable names are case-sensitive (\$age and \$AGE are two different variables)

Remember that PHP variable names are case-sensitive!

PHP Variables

Output Variables

The PHP `echo` statement is often used to output data to the screen.

- › The following example will show how to output text and a variable:

```
<?php  
$txt = "W3Schools.com";  
echo "I love " . $txt . "!";  
?>
```

- › The following example will output the sum of two variables:

```
<?php  
$x = 5;  
$y = 4;  
echo $x + $y;  
?>
```



PHP echo and print Statements

With PHP, there are two basic ways to get output: `echo` and `print`.

PHP echo and print Statements

`echo` and `print` are more or less the same. They are both used to output data to the screen.

The differences are small:

- › `echo` has no return value while `print` has a return value of 1 so it can be used in expressions.
- › `echo` can take multiple parameters (although such usage is rare) while `print` can take one argument.
- › `echo` is marginally faster than `print`.

PHP echo and print Statements

The PHP echo Statement

The `echo` statement can be used with or without parentheses: `echo` or `echo()`.

Display Text

The following example shows how to output text with the `echo` command (notice that the text can contain HTML markup):

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```

PHP echo and print Statements

Display Variables

The following example shows how to output text and variables with the `echo` statement:

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```



PHP echo and print Statements

The PHP print Statement

The `print` statement can be used with or without parentheses:
`print` or `print()`.

Display Text

The following example shows how to output text with the `print` command (notice that the text can contain HTML markup):

```
<?php  
print "<h2>PHP is Fun!</h2>";  
print "Hello world!<br>";  
print "I'm about to learn PHP!<br>";  
?>
```



PHP echo and print Statements

Display Variables

The following example shows how to output text and variables with the `print` statement:

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>
```