

PHP Syntax

The PHP script can be embedded within HTML web pages.

Standard PHP Syntax

A PHP script starts with the `<?php` and ends with the `?>` tag.

The PHP delimiter `<?php` and `?>` in the following example simply tells the PHP engine to treat the enclosed code block as PHP code, rather than simple HTML.

Example	Run this code »
<pre><?php // Some code to be executed echo "Hello, world!"; ?></pre>	

Every PHP statement end with a semicolon (`;`) — this tells the PHP engine that the end of the current statement has been reached.

Embedding PHP within HTML

PHP files are plain text files with `.php` extension. Inside a PHP file you can write HTML like you do in regular HTML pages as well as embed PHP codes for server side execution.

Example	Run this code »
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>A Simple PHP File</title> </head> <body> <h1><?php echo "Hello, world!"; ?></h1> </body> </html></pre>	

The above example shows how you can embed PHP codes within HTML to create well-formed dynamic web pages. If you view the source code of the resulting web page in your browser, the

only difference you will see is the PHP code `<?php echo "Hello, world!"; ?>` has been replaced with the output "Hello, world!".

What happen here is? when you run this code the PHP engine executed the instructions between the `<?php ... ?>` tags and leave rest of the thing as it is. At the end the web server send the final output back to your browser which is completely in HTML.

PHP Comments

A comment is simply text that is ignored by the PHP engine. The purpose of comments is to make the code more readable. It may help other developer (or you in the future when you edit the source code) to understand what you were trying to do with the PHP.

PHP support single-line as well as multi-line comments. To write a single-line comment either start the line with either two slashes (`//`) or a hash symbol (`#`). For example:

Example	Run this code »
<pre><?php // This is a single line comment # This is also a single line comment echo "Hello, world!"; ?></pre>	

However to write multi-line comments, start the comment with a slash followed by an asterisk (`/*`) and end the comment with an asterisk followed by a slash (`*/`), like this:

Example	Run this code »
<pre><?php /* This is a multiple line comment block that spans across more than one line */ echo "Hello, world!"; ?></pre>	

Case Sensitivity in PHP

Variable names in PHP are case-sensitive. As a result the variables `$color`, `$Color` and `$COLOR` are treated as three different variables.

Example

[Run this code »](#)

```
<?php
// Assign value to variable
$color = "blue";

// Try to print variable value
echo "The color of the sky is " . $color . "<br>";
echo "The color of the sky is " . $Color . "<br>";
echo "The color of the sky is " . $COLOR . "<br>";
?>
```

If you try to run the above example code it will only display the value of the variable `$color` and produce the "Undefined variable" warning for the variable `$Color` and `$COLOR`.

However the keywords, function and classes names are case-insensitive. As a result calling the `gettype()` or `GETTYPE()` produce the same result.

Example

[Run this code »](#)

```
<?php
// Assign value to variable
$color = "blue";

// Get the type of a variable
echo gettype($color) . "<br>";
echo GETTYPE($color) . "<br>";
?>
```

If you try to run the above example code both the functions `gettype()` and `GETTYPE()` gives the same output, which is: string.