

PHP - Syntax

The syntax rules of PHP are very similar to C Language. PHP is a server side scripting language. A PHP code is stored as a text file with ".php" extension. A ".php" file is essentially a web page with one or more blocks of PHP code interspersed in the HTML script. However, it must be opened in a browser with a HTTP protocol URL. In other words, if you double-click on the PHP file icon, it will be opened locally with the file protocol. For example, if you open the "index.php" file in the document root folder of your Apache server, it may just show the text of the PHP code. However, if you launch the Apache server and open the URL **http://localhost/index.php**, it displays the Apache home page.

A ".php" file may contain HTML, CSS and JavaScript code blocks along with the PHP code. Hence, the PHP parser must differentiate between the PHP code from the other elements. When a ".php" file is opened in the web browser, the HTML engine renders the HTML/CSS/JavaScript part and escapes out of the HTML block as soon as the statements included in PHP tags are encountered. The PHP parser interpreter processes this block and returns the response to the browser.



PHP defines two methods of using tags for escaping the PHP code from HTML. Canonical PHP tags and Short-open (SGML-style) tags.

Canonical PHP Tags

The most universally effective PHP tag style is –

```
<?php
    One or more PHP statements
?>
```

If you use this style, you can be positive that your tags will always be correctly interpreted.

Short-open (SGML-style) Tags

Short or short-open tags look like this –

```
<?php
    One or more PHP statements
?>
```

Short tags are, as one might expect, the shortest option. You must do one of two things to enable PHP to recognize the tags –

- Choose the "--enable-short-tags" configuration option when you're building PHP.
- Set the "short_open_tag" setting in your php.ini file to on.

```
short_open_tag=on
```

This option must be disabled to parse XML with PHP because the same syntax is used for XML tags.

The use of **ASP-style tags** –

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

and **HTML script tags** –

```
<script language = "PHP">...</script>
```

has been discontinued.

Escaping from HTML

The PHP parser ignores everything outside of a pair of opening and closing tags. Thus, a PHP file can have mixed content. This allows PHP to be embedded in HTML documents –

```
<p>This is a HTML statement</p>
<?php echo This is a PHP statement.'; ?>
<p>This is another HTML statement.</p>
```

A little advanced example of escaping using conditions is shown below –

```
<?php if ($expression == true): ?>
    This HTML statement will be rendered.
<?php else: ?>
    Otherwise this HTML statement will be rendered.
<?php endif; ?>
```

PHP skips the blocks where the condition is not met, even though they are outside of the PHP open/close tags.

For outputting large blocks of text, dropping out of PHP parsing mode is generally more efficient than sending all of the text through echo or print.

Basic Syntax of PHP

The basic syntax of PHP is very similar to that of C and C++.

Statements are expressions terminated by semicolons

A statement in PHP is any expression that is followed by a semicolon (;). Any sequence of valid PHP statements that is enclosed by the PHP tags is a valid PHP program.

Here is a typical statement in PHP, which in this case assigns a string of characters to a variable called "\$greeting" –

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

A physical line in the text editor doesn't have any significance in a PHP code. There may be multiple semicolon-terminated statements in a single line. On the other hand, a PHP statement may spill over more than one line if required.

Expressions are combinations of tokens

The smallest building blocks of PHP are the indivisible tokens such as numbers (3.14159), strings ("two"), variables (\$two), constants (TRUE), and the special words that make up the syntax of PHP itself like "if", "else", "while", "for", and so forth.

Braces make blocks

Although statements cannot be combined like expressions, you can always put a sequence of statements anywhere a statement can go, by enclosing them in a set of curly braces.

Here, both the following statements are equivalent –

```
if (3 == 2 + 1)
    print("Good - I haven't totally lost my mind.");
if (3 == 2 + 1) {
    print("Good - I haven't totally");
    print("lost my mind.");
}
```

PHP is case sensitive

PHP is a case sensitive language. The names of various PHP identifiers such as variable, function, class, etc., are case sensitive. As a result, the variable "\$age" is not the same as "\$Age". Similarly, a function called "myfunction()" is different from another function called "MyFunction()".