

# PHP syntax

Before starting programming in any language, the basic syntax and program structure is very important. This chapter introduces you to the basic PHP syntax, including: php tags, comments, print commands, echo commands, and two print and echo commands.

Before starting programming in any language, the basic syntax and program structure is very important. This chapter introduces you to the basic PHP syntax, including: php tags, comments, print commands, echo commands, and two print and echo commands.

## "Escaping to PHP" technique

PHP Parser needs a way to distinguish PHP code from other elements in the page. The technique to do this is known as "Escaping to PHP". There are 4 ways to do this:

### Accurate PHP tags

The most popular and effective PHP tag style is:

To use this card, do one or two things to enable PHP to recognize these tags:

Choose the `--enable-short-tags` configuration option when you are building PHP.

Set `short_open_tag` settings in the `php.ini` file to `on`. This option must be disabled to parse XML with PHP, because the same syntax is used for XML tags.

### ASP-style tag

ASP-style tags mimic the tags used by Active Server Pages to outline code blocks. ASP-style tags look like this:

```
% . %>
```

To use an ASP-style tag, you will need to set configuration options in the `php.ini` file.

### HTML script tags

The script tag in HTML looks like this:

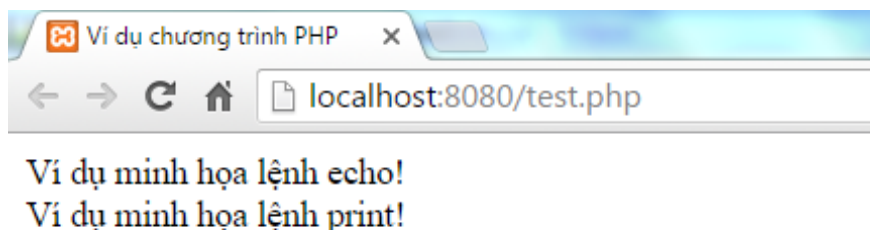
### Show text on browser - Print command and echo command

In PHP, you can use two print and echo commands to print a certain string screen. For new learners of PHP, these two commands are not much different and you can use any command.

**For example:**

```
PHP program example    php echo "Example of echo command!"  
" ; print " Example of print print! " ; ?>
```

Save the above program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:



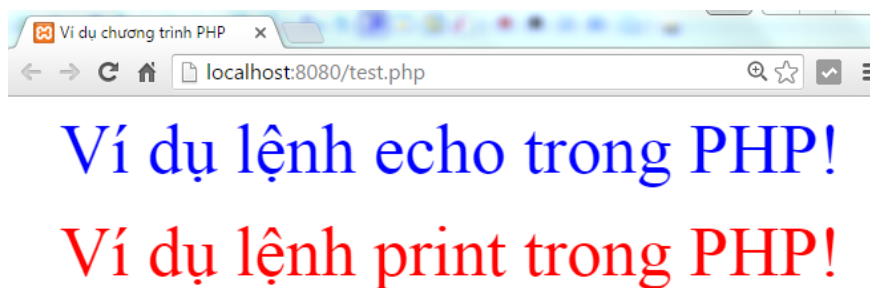
You notice the code above I use  
to down the line. If not used  
then these two lines of text will be printed together.

In other words, the print and echo commands are used to format the HTML section to display in the browser.  
You can format anything you want with the HTML in these two commands.

**For example:**

```
php echo "  
Example command echo in PHP!  
" ; //ho?c print "  
The print command example in PHP!  
" ; ?>
```

Save the program in a file named **test.php** in **htdocs** , then open the browser and type in the address **http://localhost: 8080 / test.php** will result (I minimize the browser):



You should note that the value of the attributes is enclosed in single quotes ('), but if to display quotation marks (""), you must use the forward slash (') in front. Similar to the line break (with \n), title and title with tags h1, h2, h3 .

You can also use the print command

```
php print EOF "QTM" EOF ; ?>
```

### The difference between the print command and the echo command

In PHP, basically these two commands are quite similar, but sometimes you should also pay attention to the following two different points:

The print command is a function, when executed it returns a result of 1, otherwise it returns the result 0. Therefore, you can assign the result of this print command to a variable, while with the echo command not.

```
php $QTM = print 'abcd' ; $QTM = echo 'cdef' ; //sai ?>
```

If you run the above PHP code, it will give an error.

The print command can only be used with one parameter, while the echo command can be used with many parameters.

```
php echo 'v' , 'i' , 'e' , 't' ; //dung voi 4 tham so echo ( 'j' ),( 'a' )
```

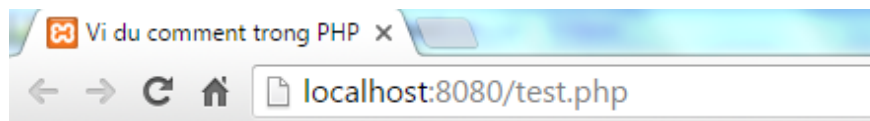
### Commet in PHP

A comment is a part of the program that only works for the code reader and it will be omitted before displaying the program results. There are 2 types of comments in PHP:

Single-line comments - In general, they are often used for brief explanations or notes regarding internal code. Below is an example of a single-line comment:

```
php # Day la vi du mot comment # Vi du comment tiep theo // Vi du ve comment
```

Save the above program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:



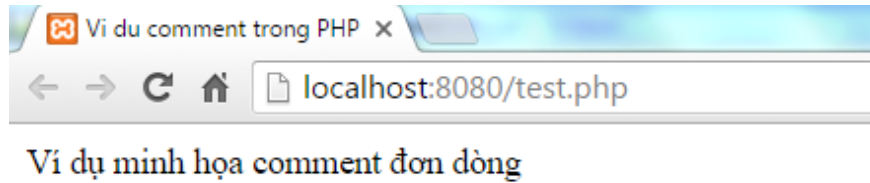
Vi dụ minh họa comment đơn dòng

Multi-line comment - Used to explain details when needed. This comment form is quite similar in C. This is an example of a multi-line comment.

```
php /* Ví dụ m?
t comment đa dòng: Web: TipsMake.com Mục đích: minh họa comment viết trên nhiều dòng
```

```
? minh h?a comment ??n dòng" ; ?>
```

Save the above program in a file named `test.php` in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:



## PHP is irrespective of whitespace

Whitespace is what you type without displaying on the screen such as: space, tab mark, or line break (end of a character line).

PHP does not distinguish whitespace, that is, there is no problem about how many whitespace characters you have in a row. A whitespace character is similar to many whitespace characters.

For example, the following PHP commands will assign a sum of `2 + 2` to the equivalent variable `$four`:

```
$four = 2 + 2 ; // cac khoang trang don $four = tab2 + tab > 2 ; // khoang t
```

## PHP is case sensitive

PHP is a case sensitive language. You consider the following example:

```
php $vietjack = 98 ; print ( "Giá tr? bi?n QTM la: $QTM  
" ); print ( "Giá tr? bi?n QTM la: $QTM  
" ); ?>
```

Save the program in a file named `test.php` in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will produce the result.

## PHP commands are expressions that are terminated by semicolons (;)

A command in PHP is any expression that is followed by a semicolon (;). Any valid PHP command sequence that is surrounded by PHP tags is a valid PHP program. Below is a typical PHP command that will assign a string to a variable of `$greeting`:

```
$ greeting = "Chasing the PHP elephant!";
```

## Expressions are combinations of tokens in PHP

The smallest blocks in PHP are tokens (which are indivisible), for example: numbers (3.14159), strings (.two.), Variables (`$two`), constants (TRUE), and special words that make up PHP's own syntax such as **if**, **else**, **while**, **for** .

## Parentheses create PHP blocks

Although commands cannot be combined like expressions, you can always place a sequence of commands surrounded by parentheses anywhere to create PHP blocks.

The following PHP commands are equivalent:

```
if ( 3 == 2 + 1 ) print ( "Hoc PHP tai QTM.  
" ); if ( 3 == 2 + 1 ) { print ( "Hoc PHP" ); print ( " tat ca co tai QTM.  
" ); }
```

### **Run the PHP script from the command prompt**

Of course, you can run your PHP script from the command prompt (command prompt). You consider the following content in the test.php file.

```
php echo "Xin chao World - Xin chao PHP!!!!!" ; ?>
```

Now run this script as a command prompt as follows:

```
$ php test.php
```

It will result:

```
Hello World - Xin chao PHP !!!!!
```

Come here, hope you understand the basics of basic PHP syntax.

### **Follow tutorialspoint**

Next article: Turning in PHP

You finished reading the article "**PHP syntax**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.