

Home
Announcements
Modules
Tasks, Tests and Surveys
Discussion and Private Messages
Gradebook
Site Info

users present:

Agnes Berthillier
Rick Boehlke
Tammy Li
Mons Osterheim
Martha Raup
Claire Rigodanzo

MODULES

[View](#) [Preferences](#)[Prev](#) | [Table Of Contents](#) | [Next](#)Week 1 » [Basics of PHP Scripts](#)

Week 1

Basics of PHP Scripts

Please check out:

<http://video.about.com/php/Basic-PHP-Syntax.htm>

Basic concepts

PHP is a classic programming language; this means that it includes variables, conditional statements, loops, functions, and so on. It is closer to JavaScript or C, for those who know these languages.

But unlike Java or JavaScript that are run using the web browser, PHP is run on the server, which allows us to do things that are not possible with HTML. The PHP program is run on the server and the result is sent to the web browser. The result is usually an HTML page.

Due to the fact that PHP is a server side scripting language, having a supporting browser is not a requirement. However, for your PHP pages to work, the hosting server must support PHP.

Our first PHP

The advantage that PHP offers over other programming languages that run on the server (such as script CGI Perl) is that you can embed PHP sentences in the HTML pages - a complicated concept if you have never studied the process of how a PHP page is processed.

Here is a simple example for a better understanding. The HTML code is in blue and the PHP code in red. We will see this in the next lesson.

`<!-- PHP Tutorial WebEstilo.com -->``<html>``<head>``<title>PHP Example</title>``</head>``<body>``HTML part``

``<?php``echo "PHP part
";``?>``</body>``</html>`

Output:

HTML part

PHP part

The PHP code runs and prints "PHP part"

When given a program to run, the PHP interpreter pays attention only to the parts of the program between PHP tags. Those tags are printed with no modification. This makes it easy to embed small bits of PHP in pages that mostly contain HTML. The commands between `<?php` (the PHP start tag) and `?>` (the PHP end tag). PHP pages typically live in files whose names end in `.php`. Below shows a page with one PHP command.

Example: Hello, World!

`<html>``<head><title>PHP says hello</title></head>``<body>````<?php``print "Hello, World!";`

```
?>
</b>
</body>
</html>
```

Basic PHP start and end tags:

Opening Tag	Closing Tag
<?php	?>
<?	?>
<script language="php">	</script>

The echo Statement and print() Function

Here are some websites that talk about echo vs print:

<http://www.webmaster-talk.com/php-forum/26613-print-vs-echo.html>

<http://codingforums.com/archive/index.php?t-41745.html>

<http://php.about.com/b/2006/06/01/to-print-or-to-echo.htm>

<http://redgem.blogspot.com/2006/11/php-print-vs-echo.html>

The only reason to use printf() in preference over echo or print() is if you will be using the format string place-holder for each such place-holder). If not, then print() will be faster, and echo even (very slightly) faster since it does not ge

Print returns its success as a boolean. Echo on the other hand does not return anything. This return call is the differen cost some cpu time but it is usually a negligible amount.

<http://www.nutt.net/2004/12/09/php-benchmarking-echo-vs-print-vs-printf/>

http://www.faqs.com/knowledge_base/view.phtml/aid/1/fid/40

<http://us2.php.net/printf>

Adding Comments to PHP Codes:

How and Why to Comment Your PHP Code

Why You Should Comment Your PHP Code, and How To Do It

A comment in PHP code is a line that is not read as part of the program. Its only purpose is to be read by so why use comments?

1. To let others know what you're doing. If you are working with a group of people, or plan script, the comments let the other programmers understand what you were doing in each step. T work with, and to edit if needed.
2. To remind yourself what you did. Although at the time you may just be writing a quick scri need for commenting, it is a good idea to add them in. Most programmers have experienced con year or two later and having to re-figure out what they did. Comments can remind you of what yo de-code your own workings.

So how do you add comments into your PHP code? There are in fact several ways to add a comment. The line. Here is an example:

```
<?php
echo "hello";
//this is a comment
echo " there";
?>
```

If you have a single line comment, another option is to use a # sign. Here is an example of this:

```
<?php
echo "hello";
#this is a comment
echo " there";
?>
```

If you have a longer, multi line comment, the best way to comment is by using /* */. You can contain several Here is an example:

```
<?php
echo "hello";
/*
```

Using this method
you can create a larger block of text
and it will all be commented out

```
*/
echo " there";
?>
```

Variables

A variable is an information container in which we can insert integers, decimals, characters... the content in and can be changed while running a PHP page.

In PHP all variables begin with the symbol \$ and it is not necessary to define a variable before using it. This means that one variable may contain a number and then characters

```
<!-- PHP Tutorial WebEstilo.com -->
<html>
<head>
  <title>PHP Example</title>
</head>
<body>
<?php
  $a = 1;
  $b = 3.34;
  $c = "Hello World";
  echo $a,"<br>",$b,"<br>",$c;
?>
</body>
</html>
```

Output:

```
1
3.34
Hello World
```

In this example, we have defined three variables, \$a, \$b and \$c, and with the echo statement, we have printed inserting a line break between them.

There are 2 kinds of variables, local variables that can only be used inside the functions, and global variables. We can access a global variable from a function using the instruction global variable_name;

Arithmetic Operators

PHP operators are very similar to C and JavaScript operators, if you know these languages, operators will be familiar to you.

These are the operators that can be applied to variables and numeric constants.

Operator	Name	Example	Description
+	Addition	5 + 6	Adds two numbers
-	Subtraction	7 - 9	Subtracts two numbers
*	Multiplication	6 * 3	Multiplies two numbers
/	Division	4 / 8	Divides two numbers
%	Modulus	7 % 2	Returns the remainder from dividing both numbers, in this case 1
++	Increment	\$a++	Adds 1 to the content of a variable.
--	Decrement	\$a--	Subtracts 1 to the content of a variable.

```
<!-- PHP Tutorial WebEstilo.com -->
<html>
<head>
  <title>PHP Example</title>
</head>
<body>
<?php
  $a = 8;
  $b = 3;
  echo $a + $b,"<br>";
  echo $a - $b,"<br>";
  echo $a * $b,"<br>";
  echo $a / $b,"<br>";
  $a++;
  echo $a,"<br>";
  $b--;
  echo $b,"<br>";
?>
</body>
```

</html>

Output:

11
5
24
2.66666666667
9
2

Comparison Operators

Comparison operators are used to compare values in order to make decisions.

Operator	Name	Example	Returns true when:
==	Equal	\$a == \$b	\$a is equal to \$b
!=	Not equal	\$a != \$b	\$a is not equal to \$b
<	Less than	\$a < \$b	\$a is less than \$b
>	Greater than	\$a > \$b	\$a is greater than \$b
<=	Less or equal	\$a <= \$b	\$a is less than or equal to \$b
>=	Greater or equal	\$a >= \$b	\$a is greater than or equal to \$b

```

<!-- PHP Tutorial WebEstilo.com -->
<html>
<head>
  <title>PHP Example</title>
</head>
<body>
<?php
  $a = 8;
  $b = 3;
  $c = 3;
  echo $a == $b,"<br>";
  echo $a != $b,"<br>";
  echo $a < $b,"<br>";
  echo $a > $b,"<br>";
  echo $a >= $c,"<br>";
  echo $b <= $c,"<br>";
?>
</body>
</html>

```

Output:

1

1
1
1

Logical Operators

Logical operators are used to evaluate several comparisons, by combining their possible values.

Operator	Name	Example	Returns true when:
&&	Y	(7>2) && (2<4)	Returns true when both conditions are true.
and	Y	(7>2) and (2<4)	Returns true when both conditions are true.
	O	(7>2) (2<4)	Returns true when at least one of the two conditions is true.
or	O	(7>2) or (2<4)	Returns true when at least one of the two conditions is true.
!	No	!(7>2)	Denies the value of the expression.

```

<!-- PHP Tutorial WebEstilo.com -->
<html>
<head>
  <title>PHP Example</title>
</head>
<body>
<?php
  $a = 8;
  $b = 3;
  $c = 3;
  echo ($a == $b) && ($c > $b),"<br>";
  echo ($a == $b) || ($b == $c),"<br>";
  echo !($b <= $c),"<br>";
?>
</body>
</html>

```

Output:

1