
Cyber Aces

Module 3 – System Administration

Web Scripting - PHP Syntax

By Tom Hessman & Michael Coppola
Presented by Tim Medin
v15Q1

This tutorial is licensed for personal use exclusively for students and teachers to prepare for the Cyber Aces competition. You may not use any part of it in any printed or electronic form for other purposes, nor are you allowed to redistribute it without prior written consent from the SANS Institute.

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

1

Welcome to Cyber Aces, Module 3! This module provides an introduction to the Apache Web Server, HTML, PHP, and basic web security.

Course Roadmap

- Introduction
- Apache & HTML
- **PHP**
- Basic Web Security
- Conclusion

- PHP Syntax
- Variables
- Echo Statement
- Strings
- Math Operators
- Comparison Operators
- Logical Operators
- If...Else
- While loops
- For loops
- Using PHP from the CLI
- Include files

Course Roadmap

In this section, we will learn about PHP, including how to install and configure it, and how to use its basic functionality.

Scripting in PHP

- PHP is a commonly used scripting language for generating dynamic web pages
- PHP powers many of the Internet's most popular sites, including Facebook and Wikipedia
- It was designed to be easy to use, allowing for rapid development
- It also has full CLI (command line interface) support



Scripting in PHP

PHP (a recursive acronym for "PHP: Hypertext Processor") is a commonly used scripting language that was designed to generate dynamic web pages. PHP is used to power many of the Internet's most popular sites, including Facebook and Wikipedia. It was designed to be easy to use and allow for rapid development. PHP also features full command line interface (CLI) support, allowing it to be used to help with system administration tasks.

Introduction to Programming in PHP

- PHP is a powerful language that can be used for many purposes
- In addition to being used to design dynamic web content, it can also be used for CLI scripting
- This section will provide you a basic overview of various key aspects of PHP, such as its syntax, variables, conditional statements, and loops
- To learn how to install a local PHP environment, see lesson 6.

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

4

Introduction to Programming in PHP

PHP is a powerful language that can be used for many purposes. This course will provide you a basic overview of PHP covering various key aspects of PHP, such as its syntax, variables, conditional statements, and loops.

To learn how to install a local PHP environment, see lesson 6.

PHP Syntax

- PHP code and HTML are mixed together in the same file, with a ".php" extension
- PHP code should begin with "<?php" and end with "?>"
 - PHP has a non-default option to begin code with "<?", but you should not rely on that being enabled on all servers
 - You do not have to place "?>" at the end of a file if no HTML follows it
- Every line should end with a semicolon (";"), except for the first line of a statement (such as "if" and "while")
- PHP files are plain ASCII text, and should be edited with a plain text editor or an IDE, NOT with word processing software
- PHP ignores extra whitespace between statements

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

5

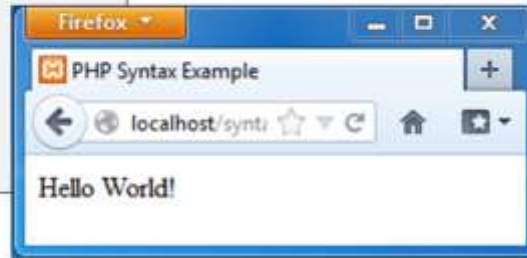
PHP Syntax

PHP code and HTML are mixed together in the same file, which should have a file extension of ".php". PHP code should begin with "<?php" and end with "?>". Depending on the server's configuration, you can also use "<?", but that feature is not enabled by default. You can have multiple blocks of PHP code within the same file, with regular HTML in between. However, you do not have to place "?>" at the end of every file. Every line of PHP code should end with a semicolon (;), except for the first line of a statement (such as an "if" statement and loops). PHP files are plain ASCII text, and should be edited using a plain text editor or an integrated development environment (IDE), NOT with word processing software. PHP also ignores extra whitespace between statements, allowing you to space code out to make it more readable.

Further reading: <http://www.tizag.com/phpT/syntax.php>

PHP Syntax Example

```
<html>
<head>
<title>PHP Syntax Example</title>
</head>
<body>
<?php
echo "Hello World!";
?>
</body>
</html>
```



Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

6

PHP Syntax Example

This simple example demonstrates basic PHP syntax. The first five lines are a normal HTML document. The sixth line, "<?php", marks the beginning of PHP code. The seventh line contains a PHP statement that sends the text "Hello World" to the browser. Note that the line ends with a semicolon! Then, the eighth line contains "?>", marking the end of PHP code. Everything after that is regular HTML again.

The screenshot of Firefox shows how that page looks when the web page is accessed in a browser. Since the PHP code is interpreted by the server, it is not sent to the client at all. If the client were to use the browser's "View Source" feature, they would only see the HTML and output generated by PHP, not the PHP itself. The user would see the following when viewing the source of the web page in the browser:

```
<html>
<head>
<title>PHP Syntax Example</title>
</head>
<body>
Hello World!
</body>
</html>
```

Note that it is also possible to have a PHP file that starts with "<?php" and does not contain any separate HTML code.

Review

- Which of the following opening tags will work on all PHP installations?
 - A) `<?php`
 - B) `<?`
 - C) `<%`
 - A & B
- Which character do you have to place at the end of every PHP statement?
 - Period (.)
 - Colon (:)
 - Semicolon (;)
 - Exclamation mark (!)

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

7

Review

Which of the following opening tags will work on all PHP installations?

- A) **`<?php`**
- B) `<?`
- C) `<%`
- D) Both A & B

Which character do you have to place at the end of every PHP statement?

- Period (.)
- Colon (:)
- Semicolon (;)
- Exclamation mark (!)

Answers

- Which of the following opening tags will work on all PHP installations?
A) `<?php`
The "`<?`" opening tag will only work if "short_open_tag" is set to "On" in php.ini, but it is off by default
- Which character do you have to place at the end of every PHP statement?
Answer: Semicolon(`;`)
All statements must end with a semicolon, just like in C/C++, which allows for flexible whitespace

Which of the following opening tags will work on all PHP installations?

A) `<?php`

The "`<?`" opening tag will only work if "short_open_tag" is set to "On" in php.ini, but it is off by default

Which character do you have to place at the end of every PHP statement?

Answer: Semicolon(`;`)

All statements must end with a semicolon, just like in C/C++, which allows for flexible whitespace

Variables

- A variable is a means of storing information for later use
- Variables allow for the manipulation of data, as well as the repetitive use of data
 - They are particularly important in PHP for storing and manipulating user input
- In PHP, all variable names start with a dollar sign (\$), and can contain letters, numbers, and underscores (but cannot start with a number)
- Variables are assigned using "=", the assignment operator

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

9

Variables

A variable is a means of storing information for later use. Variables allow for the manipulation of data, and are particularly important in PHP programming for storing and manipulating user input. In PHP, all variable names start with a dollar sign (\$), and can contain letters, numbers, and underscores (though they cannot start with a number). Variables are assigned using "=", the assignment operator ("=" is not used for testing if something is equal to something else...we'll cover that in a few slides).

Further reading: <http://www.tizag.com/phpT/variable.php>

Variable Example

```
<?php
$string = "Hello World";
$line_break = "<br>";
$number42 = 42;

echo $string;
echo $line_break;
echo $number42;
```



Variable Example

This simple example demonstrates assigning and displaying variables. The second line defines a variable called "string", and sets it to the text "Hello World". The third line defines a variable called "line_break", and sets it equal to the text "
" (an HTML line break). The fourth line defines a variable called "number42", and sets it to the number "42". The last three lines display the three variables to the browser. Note that "
" is interpreted as a line break in HTML, and is not displayed literally.

Review

- Which of the following characters must be at the beginning of all variable names?
 - Ampersand (&)
 - Percent sign (%)
 - Dollar sign (\$)
 - At Sign (@)
- Which of the following variable names is invalid?
 - \$MyVariable
 - \$_myVariable
 - \$everafter
 - \$4ever

Review

Which of the following characters must be at the beginning of all variable names?

- Ampersand (&)
- Percent sign (%)
- Dollar sign (\$)
- At Sign (@)

Which of the following variable names is invalid?

- \$MyVariable
- \$_myVariable
- \$everafter
- \$4ever

Answers

- Which of the following characters must be at the beginning of all variable names?

Answer: Dollar Sign (\$)

- Which of the following variable names is invalid?

\$4ever

Variables names cannot start with a number (though they can contain numbers after the first character)

Answers

Which of the following characters must be at the beginning of all variable names?

Answer: Dollar Sign (\$)

Which of the following variable names is invalid?

\$4ever

Variables names cannot start with a number (though they can contain numbers after the first character)

Echo

- "echo" is used to output or display information to the user, such as:
 - A string of text
 - Results of a mathematical operation
 - Contents of a variable
- When used with double-quotes (standard quotation marks), PHP will expand variables within the string to their value
 - To echo a quotation mark, either use single quotes instead of double quotes, or precede it with a backslash (\)
- Multiple echo statements will echo on the same line, unless you echo a line break character (\n) or an HTML line break (
)

Echo

PHP's "echo" command is one of the most used language constructs. It allows you to output or display information to the user, such as a string of text, the results of a mathematical operation, or a variable. When used with double-quotes (standard quotation marks), PHP will expand variables within the string to their value. To echo a quotation mark, either use single quotes instead of double quotes, or precede it with a backslash (\) to escape it (more on that later).

Further reading: <http://www.tizag.com/phpT/echo.php>

Echo Example

```
<?php
$hello = "Hello, world!";
$number = 42;
$other_number = 5;

echo $hello;
echo "<br>";
echo "A number: $number<br>";
echo $number + $other_number;
```



Echo Example

This simple example demonstrates a few common users of the echo statement. The first echo statement displays the contents of the variable named "hello". The second echo statement displays the literal string "
", which is an HTML line break. The third echo statement displays a literal string that contains a variable within it, which is expanded to its value since the string is enclosed in double quotes. The final echo statement displays the results of adding two variables together (mathematical addition, not string concatenation).

Review

- What will the following code snippet display to the user?

```
<?php
$name = "Chuck";
echo "Hello, $name!";
```

 - a. Hello, \$name!
 - b. Hello, \$Chuck!
 - c. Hello, Chuck!
 - d. Nothing, because the code snippet is invalid.
- What will the following code snippet display to the user?

```
<?php
$name = "Charles";
$quote = "Aces, $name";
echo "Chuck's dad always used to say \"$quote\" to him as a kid.";
```

 - a. Chuck's dad always used to say \"Aces, \$name\" to him as a kid.
 - b. Chuck's dad always used to say "Aces, Charles" to him as a kid.
 - c. Chuck's dad always used to say \"Aces, Charles\" to him as a kid.
 - d. Nothing, because the code snippet is invalid.

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

15

Review

What will the following code snippet display to the user?

```
<?php
$name = "Chuck";
echo "Hello, $name!";
```

- a. Hello, \$name!
- b. Hello, \$Chuck!
- c. Hello, Chuck!
- d. Nothing, because the code snippet is invalid.

What will the following code snippet display to the user?

```
<?php
$name = "Charles";
$quote = "Aces, $name";
echo "Chuck's dad always used to say \"$quote\" to him as a kid.";
```

- a. Chuck's dad always used to say \"Aces, \$name\" to him as a kid.
- b. Chuck's dad always used to say "Aces, Charles" to him as a kid.
- c. Chuck's dad always used to say \"Aces, Charles\" to him as a kid.
- d. Nothing, because the code snippet is invalid.

Answers

- What will the following code snippet display to the user?

```
<?php
$name = "Chuck";
echo "Hello, $name!";
```

The answer is C: Hello, Chuck!

 - The variable "\$name" gets expanded to its value.
- What will the following code snippet display to the user?

```
<?php
$name = "Charles";
$quote = "Aces, $name";
echo "Chuck's dad always used to say \"$quote\" to him as a kid.";
```

The answer is B

Chuck's dad always used to say "Aces, Charles" to him as a kid.

The variable "\$name" gets expanded to its value within "\$quote", and "\$quote" gets expanded to its value within the echo statement. Also, the backslashes indicate that the quotation mark should be displayed literally, and not be interpreted as the end of the string.

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

16

Answers

What will the following code snippet display to the user?

```
<?php
```

```
$name = "Chuck";
```

```
echo "Hello, $name!";
```

Answer is C: Hello, Chuck!

The variable "\$name" gets expanded to its value.

What will the following code snippet display to the user?

```
<?php
```

```
$name = "Charles";
```

```
$quote = "Aces, $name";
```

```
echo "Chuck's dad always used to say \"$quote\" to him as a kid.";
```

Answer is B:

Chuck's dad always used to say "Aces, Charles" to him as a kid.

The variable "\$name" gets expanded to its value within "\$quote", and "\$quote" gets expanded to its value within the echo statement. Also, the backslashes indicate that the quotation mark should be displayed literally, and not be interpreted as the end of the string.

Strings

- In programming, a "string" is a series of single characters (such as 'A' and 'B') that are strung together as a single unit
- They are used to store arbitrary text, and are therefore very important for processing user data
- Strings can be enclosed in either single quotes (') or double quotes (")
 - With single quotes, everything between the quotes is interpreted literally, except for a backslash followed by a single quote (which inserts a literal single quote).
 - With double quotes, special characters are interpreted within the string, such as "\n" (newline character) and "\t" (a tab character), plus variables are expanded to their values

Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

17

Strings

In programming, a string is a series of single characters (such as 'A' and 'B') that are strung together as a single unit. Strings are used to store arbitrary text, and therefore are very important for processing user data.

Strings can be enclosed in either single quotes (') or double quotes ("). When a string is enclosed in single quotes, everything between the quotes is interpreted literally, except for a backslash followed by a single quote (which inserts a literal single quote). When a string is enclosed in double quotes, special characters are interpreted, such as "\n" (a newline character) and "\t" (a tab character), plus variables are expanded to their values.

Further reading: <http://www.tizag.com/phpT/strings.php>

String Example

```
<?php
$agentP = "Perry";

echo $agentP;
echo '<br />';
echo 'Where\'s $agentP?';
echo "<br />";
echo "Where's $agentP?";
```



Cyber Aces Module 3 - ©2015 The SANS Institute. Redistribution Prohibited.

18

String Example

This example shows the different ways that strings can be used with single and double quotes. The first echo statement, **echo \$agentP;**, simply displays the contents of the variable called "\$agentP" to the browser. The third echo statement, **echo 'Where\'s \$agentP?';**, displays the literal string contained within single quotes, except that it replaces \' with just '. That's why it displays the literal text "\$agentP" instead of the contents of the variable "\$agentP". The final echo statement, using double quotes, doesn't need to escape the single quote, and it does expand the variable "\$agentP" to its value.

Review

- How can a newline character be represented in a string?

\t
'
\n
\\

- What will the following code snippet display to the user?

```
<?php
$name = 'Chuck';
echo 'Hello, $name!';
```

a. Hello, Chuck!
b. Hello, \$name!
c. Hello, Chuck
d. Nothing, because the code snippet is invalid.

How can a newline character be represented in a string?

\t
'
\n
\\

What will the following code snippet display to the user?

```
<?php
$name = 'Chuck';
echo 'Hello, $name!';
```

a. Hello, Chuck!
b. Hello, \$name!
c. Hello, Chuck
d. Nothing, because the code snippet is invalid.

Answers

- How can a newline character be represented in a string?

`\n`

This represents a literal newline character (like hitting the "Enter" key in a text document). It does *not* cause a newline to appear in HTML, which requires the HTML "
" tag (though it will cause a newline to appear in the HTML source).

- What will the following code snippet display to the user?

```
<?php
$name = 'Chuck';
echo 'Hello, $name!';
```

The answer is B: Hello, \$name!

Because the string is contained in single quotes, the dollar sign (\$) is interpreted literally instead of expanding the variable \$name.

Answers

How can a newline character be represented in a string?

`\n`

This represents a literal newline character (like hitting the "Enter" key in a text document). It does not cause a newline to appear in HTML, which requires the HTML "
" tag (though it will cause a newline to appear in the HTML source).

What will the following code snippet display to the user?

```
<?php
$name = 'Chuck';
echo 'Hello, $name!';
```

The answer is B: Hello, \$name!

Because the string is contained in single quotes, the dollar sign (\$) is interpreted literally instead of expanding the variable \$name.

Tutorial Complete!

- This concludes the introduction to PHP and its syntax
- Next, you'll learn about PHP Operators

Tutorial Complete

This concludes the first section of Module 3. We've learned the basic syntax of PHP. Next, we'll learn about PHP's operators.