Lesson Proper for Week 15

Hypertext Preprocessor (PHP)

PHP is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).

PHP was created by **Rasmus Lerdorf in 1994** but appeared in the market in 1995. **PHP 7.4.0** is the latest version of PHP, which was released on **28 November**. Some important points need to be noticed about PHP are as followed:

¦ PHP stands for Hypertext Preprocessor.

¦ PHP is an interpreted language, i.e., there is no need for compilation.

¦ PHP is faster than other scripting languages, for example, ASP and JSP.

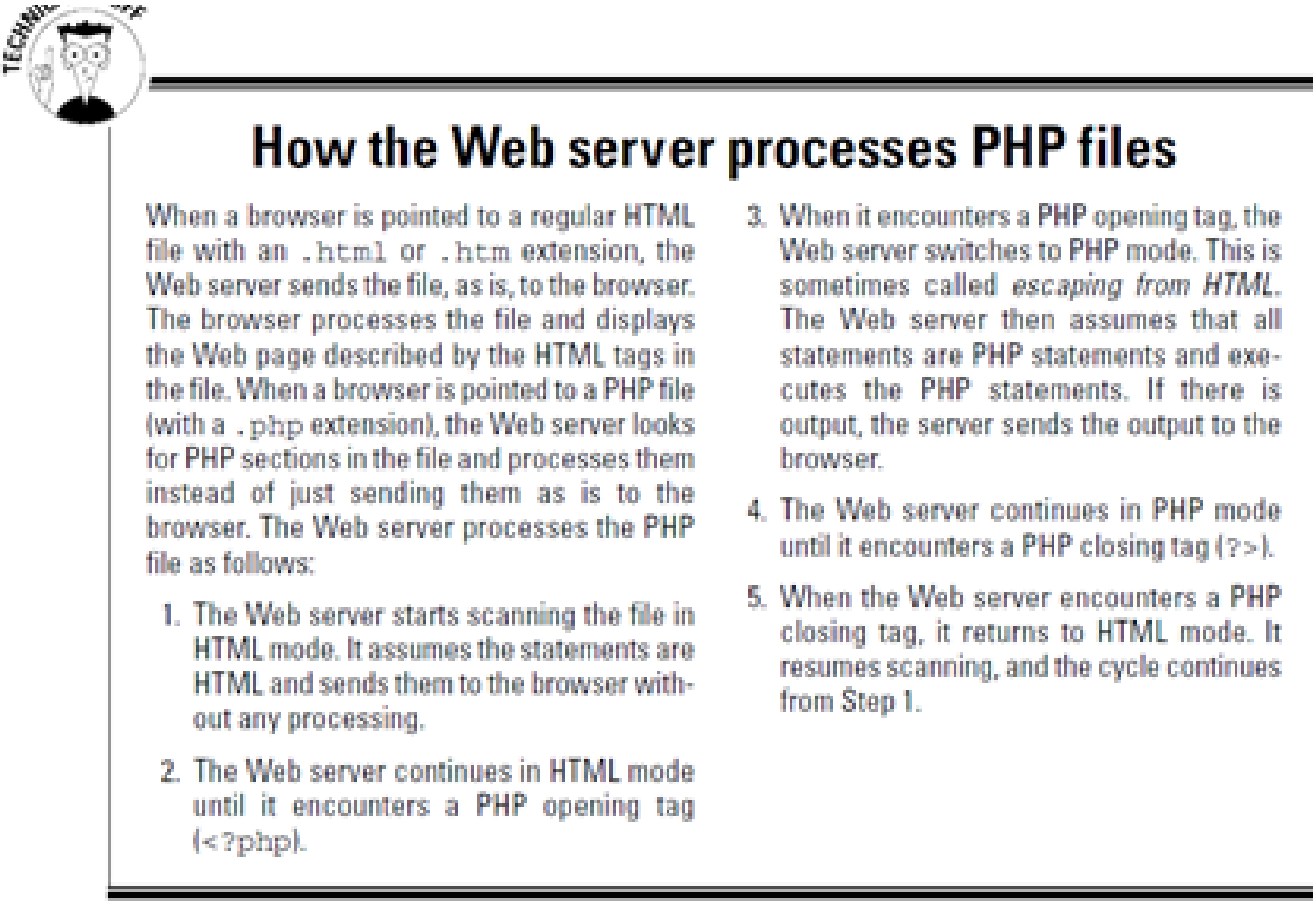
¦ PHP is a server-side scripting language, which is used to manage the dynamic content of the website.

¦ PHP can be embedded into HTML.

¦ PHP is an object-oriented language.

¦ PHP is an open-source scripting language.

¦ PHP is simple and easy to learn language.



How the Web server processes PHP files

When a browser is pointed to a regular HTML file with an `.html` or `.htm` extension, the Web server sends the file, as is, to the browser. The browser processes the file and displays the Web page described by the HTML tags in the file. When a browser is pointed to a PHP file (with a `.php` extension), the Web server looks for PHP sections in the file and processes them instead of just sending them as is to the browser. The Web server processes the PHP file as follows:

1. The Web server starts scanning the file in HTML mode. It assumes the statements are HTML and sends them to the browser without any processing.
2. The Web server continues in HTML mode until it encounters a PHP opening tag (`<?php`).
3. When it encounters a PHP opening tag, the Web server switches to PHP mode. This is sometimes called *escaping from HTML*. The Web server then assumes that all statements are PHP statements and executes the PHP statements. If there is output, the server sends the output to the browser.
4. The Web server continues in PHP mode until it encounters a PHP closing tag (`?>`).
5. When the Web server encounters a PHP closing tag, it returns to HTML mode. It resumes scanning, and the cycle continues from Step 1.

How PHP Works

The PHP software works with the *Web server.* The Web server is the software that delivers Web pages to the world. When you type a URL into your Web browser's address bar, you're sending a message to the Web server at that URL, asking it to send you an HTML file. The Web server responds by sending the requested file. Your browser reads the HTML file and displays the Web page. You also request a file from the Web server when you click a link in a Web page. In addition, the Web server processes a file when you click a Web page button that submits a form.

Structure of a PHP Script

PHP is an *embedded* scripting language when used in Web pages. This means that PHP code is embedded in HTML code. You use HTML tags to enclose the PHP language that you embed in your HTML file, the same way that you would use other HTML tags. You create and edit Web pages containing PHP the same way that you create and edit regular HTML pages.

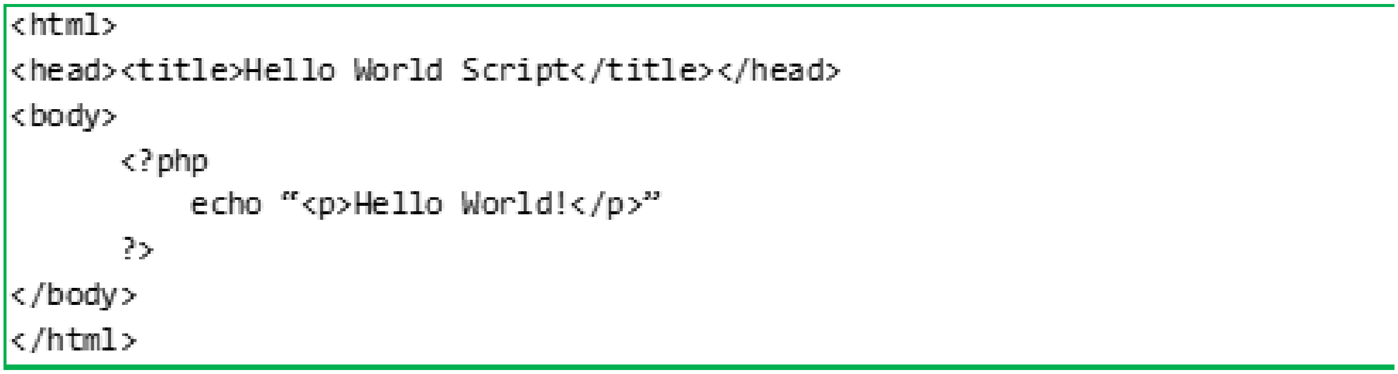
The PHP language statements are enclosed in PHP tags with the following form:

```
<?php
        statements……. .
?>
```

PHP processes all statements between the two PHP tags. After the PHP section is processed, it's discarded. Or if the

PHP processes all statements between the two PHP tags. After the PHP section is processed, it's discarded. Or if the PHP statements produce output, the PHP section is replaced by the output. The browser doesn't see the PHP section — the browser sees only its output, if there is any. For more on this process, see the sidebar "How the Web server processes PHP files.

Example:



```
<html>
<head><title>Hello World Script</title></head>
<body>
      <?php
          echo "<p>Hello World!</p>"
      ?>
</body>
</html>
```

When you run this script, by looking at it in your browser, it displays the same Web page as the HTML script.

The PHP tags enclose only one statement — an **echo** statement. The **echo** statement is a PHP statement that you'll use frequently. The output is simply the text that's included between the double quotes.

PHP Syntax

The PHP section that you add to your HTML file consists of a series of PHP statements. Each PHP statement is an instruction to PHP to do something. PHP statements can be simple or complex.

Using simple statements

Simple statements are an instruction to PHP to do one simple action. The echo statement is a simple PHP statement that instructs PHP to output the text between the double quotes. PHP simple statements follow these rules:

¦ **PHP statements end with a semicolon (;) or the PHP ending tag.**

PHP doesn't notice white space or the end of lines. It continues reading a statement until it encounters a semicolon or the PHP closing tag, no matter how many lines the statement spans.

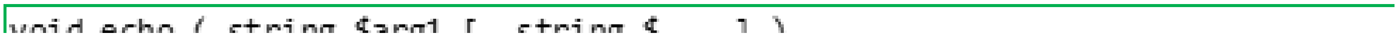
¦ **PHP statements may be written in either upper- or lowercase.**

In an **echo** statement, Echo, echo, **ECHO**, and eCHo are all the same to PHP.

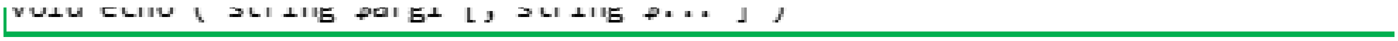
PHP Echo

PHP echo is a language construct, not a function. Therefore, you don't need to use parenthesis with it. But if you want to use more than one parameter, it is required to use parenthesis.

The syntax of PHP echo is given below:



```
void echo ( string $arg1 [ , string $... ] )
```



```
void echo ( string $arg1 [, string $... ] )
```

PHP echo statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

¦ echo is a statement, which is used to display the output.

¦ echo can be used with or without parentheses: echo(), and echo.

¦ echo does not return any value.

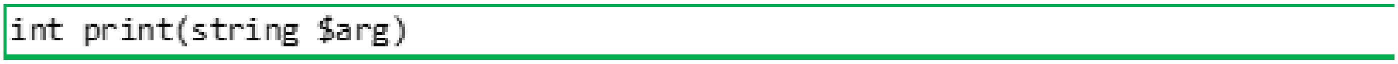
¦ We can pass multiple strings separated by a comma (,) in echo.

¦ echo is faster than the print statement.

PHP Print

Like PHP echo, PHP print is a language construct, so you don't need to use parenthesis with the argument list. Print statement can be used with or without parentheses: print and print(). Unlike echo, it always returns 1.

The syntax of PHP print is given below:



```
int print(string $arg)
```

PHP print statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

¦ print is a statement, used as an alternative to echo at many times to display the output.

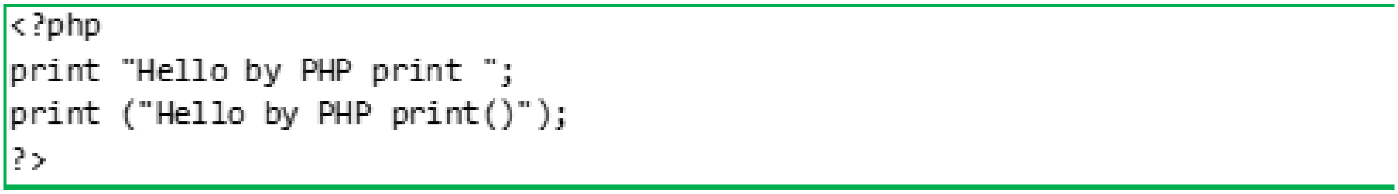
¦ print can be used with or without parentheses.

¦ print always returns an integer value, which is 1.

¦ Using print, we cannot pass multiple arguments.

¦ print is slower than the echo statement.

Example: PHP printing string:



```
<?php
print "Hello by PHP print ";
print ("Hello by PHP print()");
?>
```

Using PHP Variables

Variables are containers used to hold information. A variable has a name, and information is stored in the variable. In PHP, a variable is declared using a **$ sign** followed by the variable name. Here, some important points to know about variables:

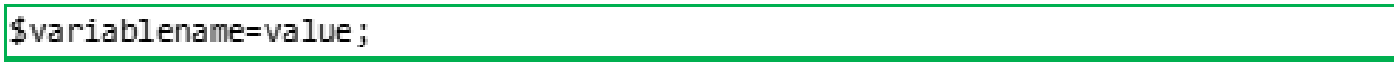
¦ As PHP is a loosely typed language, so we do not need to declare the data types of the variables. It automatically

¦ As PHP is a loosely typed language, so we do not need to declare the data types of the variables. It automatically analyzes the values and makes conversions to its correct datatype.

¦ After declaring a variable, it can be reused throughout the code.

¦ Assignment Operator (=) is used to assign the value to a variable.

Syntax of declaring a variable in PHP is given below:



```
$variablename=value;
```

Rules for declaring PHP variable:

¦ A variable must start with a dollar ($) sign, followed by the variable name.

¦ It can only contain alpha-numeric character and underscore (A-z, 0-9, _).

¦ A variable name must start with a letter or underscore (_) character.

¦ A PHP variable name cannot contain spaces.

¦ One thing to be kept in mind that the variable name cannot start with a number or special symbols.

¦ PHP variables are case-sensitive, so $name and $NAME both are treated as a different variable.

Using PHP Constants

PHP *constants* are similar to variables. Constants are given a name, and a value is stored in them. However, constants are constant; that is, they can't be changed by the script. After you set the value for a constant, it stays the same. If you used a constant for age and set it to 21, for example, it can't be changed.

Understanding Data Types

PHP data types are used to hold different types of data or values. Values stored in a variable or a constant are stored as a specific type of data. PHP supports 8 primitive data types that can be categorized further in 3 types:

1. **PHP Data Types: Scalar Types -** It holds only single value. There are 4 scalar data types in PHP.

   i. *Integer:* A whole number

Integer means numeric data with a negative or positive sign. It holds only whole numbers, i.e., numbers without fractional part or decimal points.

Rules for integer:

o An integer can be either positive or negative.

o An integer must not contain decimal point.

o Integer can be decimal (base 10), octal (base 8), or hexadecimal (base 16).

o The range of an integer must be lie between 2,147,483,648 and 2,147,483,647 i.e., -2^31 to 2^31.

   ii. *Floating-point number:* A numeric value with decimal digits

ii. *Floating point number:* A numeric value with decimal digits

A floating-point number is a number with a decimal point. Unlike integer, it can hold numbers with a fractional or decimal point, including a negative or positive sign.

iii. *String*: A series of characters

A string is a non-numeric data type. It holds letters or any alphabets, numbers, and even special characters. String values must be enclosed either within single quotes or in double-quotes. But both are treated differently.

iv. *Boolean:* A value that can be either true or false

Booleans are the simplest data type that works like a switch. It holds only two values: **TRUE (1)** or **FALSE (0)**. It is often used with conditional statements. If the condition is correct, it returns TRUE otherwise FALSE.

2. **PHP Data Types: Compound Types -** It can hold multiple values. There are 2 compound data types in PHP.

i. *Array:* A group of values in one variable

An array is a compound data type. It can store multiple values of the same data type in a single variable.

ii. *Object*: A structure created with a class

Objects are instances of user-defined classes that can store both values and functions. They must be explicitly declared.

3. **PHP Data Types: Special Types –** There are 2 special data types in PHP.

i. *Resource:* A reference that identifies a connection

Resources are not the exact data type in PHP. Basically, these are used to store some function calls or references to external PHP resources. **For example** - a database call. It is an external resource.

ii. *NULL:* A value that represents no value

Null is a special data type that has only one value: **NULL**. There is a convention of writing it in capital letters as it is case-sensitive.

PHP Operators

PHP Operator is a symbol used to perform operations on operands. In simple words, operators are used to perform operations on variables or values.

PHP Operators can be categorized in following forms:

o Arithmetic Operators - The PHP arithmetic operators are used to perform common arithmetic operations suc addition, subtraction, etc. with numeric values.

Operator	Name	Example	Explanation

Operator	Name	Example	Explanation
+	Addition	$a + $b	Sum of operands
-	Subtraction	$a - $b	Difference of operands
*	Multiplication	$a * $b	Product of operands
/	Division	$a / $b	Quotient of operands
%	Modulus	$a % $b	Remainder of operands
**	Exponentiation	$a ** $b	$a raised to the power $b

o Assignment Operators - The assignment operators are used to assign value to different variables. The basic assignment operator is "=".

Operator	**Name**	**Example**	**Explanation**
=	Assign	$a = $b	The value of right operand is assigned to the left operand.
+=	Add then Assign	$a += $b	Addition same as $a = $a + $b
-=	Subtract then Assign	$a -= $b	Subtraction same as $a = $a - $b
*=	Multiply then Assign	$a *= $b	Multiplication same as $a = $a * $b
/=	Divide then Assign (quotient)	$a /= $b	Find quotient same as $a = $a / $b
%=	Divide then Assign (remainder)	$a %= $b	Find remainder same as $a = $a % $b

o Bitwise Operators - The bitwise operators are used to perform bit-level operations on operands. These operators allow the evaluation and manipulation of specific bits within the integer.

Operator	**Name**	**Example**	**Explanation**
&	And	$a & $b	Bits that are 1 in both $a and $b are set to 1, otherwise 0.
|	Or (Inclusive or)	$a | $b	Bits that are 1 in either $a or $b are set to 1
^	Xor (Exclusive or)	$a ^ $b	Bits that are 1 in either $a or $b are set to 0.
			Bits that are 1 set to 0 and bits that are 0

~	Not	~$a	Bits that are 1 set to 0 and bits that are 0 are set to 1
<<	Shift left	$a << $b	Left shift the bits of operand $a $b steps
>>	Shift right	$a >> $b	Right shift the bits of $a operand by $b number of places

o Comparison Operators - Comparison operators allow comparing two values, such as number or string. Below the list of comparison operators are given:

Operator	**Name**	**Example**	**Explanation**
==	Equal	$a == $b	Return TRUE if $a is equal to $b
===	Identical	$a === $b	Return TRUE if $a is equal to $b, and they are of same data type
!==	Not identical	$a !== $b	Return TRUE if $a is not equal to $b, and they are not of same data type
!=	Not equal	$a != $b	Return TRUE if $a is not equal to $b
<>	Not equal	$a <> $b	Return TRUE if $a is not equal to $b
<	Less than	$a < $b	Return TRUE if $a is less than $b
>	Greater than	$a > $b	Return TRUE if $a is greater than $b
<=	Less than or equal to	$a <= $b	Return TRUE if $a is less than or equal $b
>=	Greater than or equal to	$a >= $b	Return TRUE if $a is greater than or equal $b
<=>	Spaceship	$a <=>$b	Return -1 if $a is less than $b Return 0 if $a is equal $b Return 1 if $a is greater than $b

o Incrementing/Decrementing Operators - The increment and decrement operators are used to increase and decrease the value of a variable.

Operator	**Name**	**Example**	**Explanation**
++	Increment	++$a	Increment the value of $a by one, then return $a
		$a++	Return $a, then increment the value of $a by one

		$a++	Return $a, then increment the value of $a by one
--	decrement	--$a	Decrement the value of $a by one, then return $a
		$a--	Return $a, then decrement the value of $a by one

o Logical Operators - The logical operators are used to perform bit-level operations on operands. These operators allow the evaluation and manipulation of specific bits within the integer.

Operator	Name	Example	Explanation
and	And	$a and $b	Return TRUE if both $a and $b are true
Or	Or	$a or $b	Return TRUE if either $a or $b is true
xor	Xor	$a xor $b	Return TRUE if either $ or $b is true but not both
!	Not	! $a	Return TRUE if $a is not true
&&	And	$a && $b	Return TRUE if either $a and $b are true
||	Or	$a || $b	Return TRUE if either $a or $b is true

o String Operators - The string operators are used to perform the operation on strings. There are two string operators in PHP, which are given below:

Operator	Name	Example	Explanation
.	Concatenation	$a . $b	Concatenate both $a and $b
.=	Concatenation and Assignment	$a .= $b	First concatenate $a and $b, then assign the concatenated string to $a, e.g. $a = $a . $b

o Array Operators - The array operators are used in case of array. Basically, these operators are used to compare the values of arrays.

Operator	Name	Example	Explanation
+	Union	$a + $y	Union of $a and $b
==	Equality	$a == $b	Return TRUE if $a and $b have same key/value pair
!=	Inequality	$a != $b	Return TRUE if $a is not equal to $b
===	Identity	$a === $b	Return TRUE if $a and $b have same key/value pair of same type in same order
!==	Non-Identity	$a !== $b	Return TRUE if $a is not identical to $b

!==	Non-identity	$a !== $b	Return TRUE if $a is not identical to $b
<>	Inequality	$a <> $b	Return TRUE if $a is not equal to $b

◀ Preliminary Activity for Week 15

Jump to...

Analysis, Application, and Exploration for Week 15 ▶

Navigation

Home
Dashboard
Site pages
My courses
121 - CC106
121 - BPM101 / DM103
121 - OAELEC2
121 - ITE3
121 - MUL101
121 - ITSP2B
121 - WEB101 / CCS3218
Participants
Grades
General
01 Introduction to Internet and World Wide Web
02 Full Stack Development Overview
03 The Basics of HTML
04 Mark Up Text
05 HTML Link, Image and Table
06 - Preliminary Examination
07 HTML Block, Inline Element and Forms Control
08 Introduction To Cascading Style Sheets
09 The Font Style Properties
10 The List and Table style properties
11 Advanced CSS
12 - Midterm Examination
13 JavaScript Basics
14 JavaScript Objects
15 Introduction to PHP
Preliminary Activity for Week 15
Lesson Proper for Week 15

- Analysis, Application, and Exploration for Week 15
- Generalization for Week 15
- Evaluation for Week 15
- Assignment for Week 15

16 PHP Control Statement and Function

17 PHP Array and String

Courses

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission***.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

2nd Semester Enrollment

visit www.bcp.edu.ph
Enrollment registration is now Ongoing
For 2nd Semester SY 2021 - 2022
We are accepting new students, returnees and transferees.
"Be trained to be the best,
Be linked to success"
bcp-inquire@bcp.edu.ph
(8)442-8601 | (8)518-8050

Activities

- Assignments
- Forums
- Quizzes
- Resources