

PHP 101

What is PHP?

- PHP is a widely-used scripting language intended for use in web pages to create dynamic content.
- PHP runs on a web server, which interprets the PHP scripts behind the scenes and sends the results to the client as HTML.
- Top features:
 - Solving Math Problems
 - Displaying Current Dates & Times
 - Handling Form Data
 - Sending & Receiving Cookies
 - Collecting Information About Your Website's Visitors
 - Tools to Read From & Write Data to Databases
 - Creating/Modifying Images
 - Creating Shopping Carts
 - Interacting With a Secure 3rd Party to Pay For Online Merchandise
- PHP is a free open source software already installed on many web servers today.
- PHP stands for "PHP: Hypertext Preprocessor"
- Any file containing PHP must have a ".php" file extension, similar to how HTML files, to be identified as HTML files, must have a ".htm" or ".html" file extension.

Start with PHP Code

- The PHP syntax is a set of rules that define how a program should be written. PHP is, in that sense, no different from any other language.
- All code written in PHP must be identified as PHP code. A set of tags are used to mark the beginning and end of a block of code, in between which any amount of code can be written.
- The standard opening tag is:

```
<?php
```

- The standard closing tag is:

```
?>
```

- These tags can be used to jump in and out of "PHP mode" any number of times in a PHP file, including a PHP file containing HTML elements.

```
<?php /* Code Can Go Here */ ?>
<html>
<head>
  <?php /* Code Can Go Here */ ?>
</head>
<body>
  <?php /* Code Can Go Here */ ?>
</body>
</html>
<?php /* Code Can Go Here */ ?>
```

Start with PHP Code

- Each statement made in PHP needs a method of indicating the end of the statement so that each instruction can be carried out without mix-up. The semicolon ";" is used for this purpose.

```
<?php
$variable = "Why can you never swindle a snake?";
echo $variable;
?>
```

- PHP Comments:

- Comments are never seen by anyone who cannot see the php code. Three different methods can be used to create comments:

```
<?php
// This comment only spans one line.
// It can, however, be used on as many lines as necessary.

# This comment only spans one line.
# It is not used as commonly as the previous type.

/* This comment can span one line. */

/* This comment can also span as many lines as needed.
   It is useful when commenting out large chunks of code at a
   time.
   This type of comment cannot be nested, or errors will occur.
  */
?>
```

PHP Variables

- Variables are PHP's method of storing values, or information. Once a variable is set, it can be used over and over again, saving you the work of typing in the value again and again, and allowing you to assign new values spontaneously.
- Variables are identified by a dollar sign, immediately followed by a variable name. Variable names are case-sensitive, but you can name your own variables as long as they abide by four basic rules:
 1. Variable names cannot contain spaces.
 2. Variables names can contain letters (a-z and A-Z), numbers (0-9) and underscores (_).
 3. Variable names can begin with letters or an underscore, but cannot begin with numbers.
 4. Variable names should make sense so that you can remember them later!
- Several examples of variables being assigned are below:

```
<?php
$NonSensical_Variable_Name = "I am a variable value.";
$empty_variable = "";
$eyes = "brown";
$hair = 'brown';
$age = 35;
?>
```

The Echo Statement

- PHP gives two options to choose between in order to output text to the browser. Although the two options are similar, "echo" is used far more than "print" because it is generally faster, shorter to type, and sounds cool! For the examples in this tutorial we will be using echo exclusively, so let's learn all about it.

```
<?php
$numbers = 123;
echo "Why did the lizard go on a diet?";
echo 'It weighed too much for its scales!';
echo $numbers;
?>
```

- Echo accepts both variables and strings (in single or double quotes). Variables can be "echoed" by themselves, or from inside of a double-quoted string, but putting variable names inside a single-quoted string will only output the variable name. For example:

```
<?php
$numbers = 123;
echo "There are only 10 numbers. These numbers are:
$numbers";
echo 'There are only 10 numbers. These numbers are:
$numbers';
?>
```

The Echo Statement

- HTML elements can be echoed as part of a string or variable.

```
<?php
echo '<p style="color: green;"></p>';
echo "What's this? An error message?";
?>
```

- The escape character is a backslash \.

```
<?php
echo '<p style="color: green;"></p>';
echo "<p style=\"color: green;\"></p>";
echo "What's this? No error message?";
echo 'What\'s this? No error message?';
?>
```

PHP Operators

- Operators perform operations. They are symbols, or combinations of symbols, that assign value, combine values or compare values.
- PHP operators can be broken down into several categories

- **Assignment Operators**

Very simply, assignment operators assign value. We have already used the assignment operator to assign values to variables.

Operator	Description
=	Assigns A Value

```
<?php
$variable = "abcde";
echo $variable; // Result: abcde
?>
```

- **String Operators**

String operators, as we have learned, can either link two strings together, or add one string to the end of another string.

Operator	Description
.	Concatenates (Links) Two Strings Together
.=	Adds A String Onto the End of An Existing String

```
<?php
$variable = "abcde" . "fghij";
$variable .= "klmno";
echo $variable; // Result: abcdefghijklmno
?>
```


PHP Operators

▪ Arithmetic Operators

Math! Some people like it and some people don't. Why not have fun writing a PHP script that does your math for you?

Operator	Description
+	Addition Operator (Adds Two Values Together)
-	Subtraction Operator (Subtracts One Value From Another)
*	Multiplication Operator (Multiplies Two Values)
/	Division Operator (Divides One Values From Another)
%	Modulus Operator (Determines The Remainder of a Division)

```
<?php
$addition = 5 + 5;
echo $addition; // Result: 10

$subtraction = 5 - 5;
echo $subtraction; // Result: 0

$multiplication = 5 * 5;
echo $multiplication; // Result: 25

$division = 5 / 5;
echo $division; // Result: 1

$modulus = 7 % 5;
echo $modulus; // Result: 2

echo 3 + 5 / 2; // Result: 5.5

echo (3 + 5) / 2; // Result: 4
?>
```

PHP Operators

▪ Combined Assignment Operators

Combined assignment operators are shortcuts useful when performing arithmetic operations on an existing variable's numerical value. They are similar to the concatenation assignment operator.

Operator	Description
+=	Adds Value to An Existing Variable Value
-=	Subtracts Value From Existing Variable Value
*=	Multiplies An Existing Variable Value
/=	Divides An Existing Variable Value
%=	Modulo of An Existing Variable Value And New Value

Example before combined assignment operator is used:

```
<?php
$variable = 8;
$variable = $variable + 3;
echo $variable; // Result: 11
?>
```

Example of combined assignment operator in use:

```
<?php
$variable = 8;
$variable += 3;
echo $variable; // Result: 11
?>
```

PHP Operators

▪ Incrementing/Decrementing Operators

There are shortcuts for adding (incrementing) and subtracting (decrementing) a numerical value by one.

Operator	Description
++	Incrementation Operator (Increases A Value By 1)
--	Defragmentation Operator (Decreases A Value By 1)

```
<?php
$variable = 15;
$variable++;
echo $variable; // Result: 16
$variable--;
echo $variable; // Result: 15
?>
```

▪ Comparison Operators

Operator	Description	Example
==	Is Equal To	3==2 (Will Return "False")
!=	Is Not Equal To	3!=2 (Will Return "True")
<>	Is Not Equal To	3<>2 (Will Return "True")
>	Is Greater Than	3>2 (Will Return "True")
<	Is Less Than	3<2 (Will Return "False")
>=	Is Greater Than or Equal To	3>=2 (Will Return "True")
<=	Is Less Than or Equal To	3<=2 (Will Return "False")

▪ Logical Operators

Operator	Meaning	Description
&&	And	True If Two Statements Are True, Otherwise False
	Or	True If One Statement or Another Is True, Otherwise False
!	Not	True If Statement Is False, False If Statement Is True