CMPT 210
Applications in Software
Lesson 3

Objectives:

- **How to use Variables**
- **How to use Operators**

What a Variable?:

A variable is representation or a holding spot for a value, such as 23 or the string value "red". By assigning a variable a value you can reference the variable in other places in your script.

To create a variable for a username you would have to first start out with a dollar sign ($) and then assign it with the equals sign (=) and then give it a value of "Bob". That would look like this: $username = "Bob";

We now have a variable called ***username*** with the value of ***Bob***. And don't forget the terminator (;).

PHP is a Loosely Typed Language

In PHP, a variable does not need to be declared before adding a value to it. PHP will automatically convert the variable to the correct data type, depending on its value. So you do not have to declare the type of INT, Double or String.

In a strongly typed programming language, you have to declare (define) the type and name of the variable before using it such as Java.

In PHP, the variable is declared automatically when you use it.

Naming Variables:

As in the example above you must start out your variable with the dollar sign ($). You should then use a meaningful name. Variable names cannot begin with numeric characters. Case does matter! The variables $username, $USERNAME and $UserName are all different variables.

Variable Value Types

- Integers – Whole numbers. No quotations because that would make it a STRING.
- Floating-point numbers – Numbers with decimals. No quotations because that would make it a STRING.
- Strings – Text or mixed text and numbers must be in ('single quotes') or ("Double quote") This script demonstrates each of the variable types.

```
<HTML>
<HEAD>
<TITLE>Printing Variables</TITLE>
</HEAD>
<BODY><center><H2>

<?
$intVar = 9554215464;
$floatVar = 1542.2232235;
$stringVar = "This is a string.";

echo "<P>integer: $intVar</p>";
echo "<P>float: $floatVar</p>";
echo "<P>string: $stringVar</p>";

?>
</Center></H2></BODY>
</HTML>
```

Pre-Defined Variables

There are some pre-defined variables in PHP that you can use in your scripts at any time.

- $_GET – Will provide any variable through the GET method
- $_POST – Used in the post method
- $_COOKIE – Provides any variable through the script through a cookie
- $_FILES – Used in the file uploads
- $_ENV – Provides the server environment
- $_SESSION- Contains a variable that are registered in a session

Using Constants

A constant is something that will not change for the live of the script. There are two types of constants: User defined and pre-defined that PHP always has available.

Pre-Defined Constants – No "$"

- _FILE_ name of the script being parsed
- _LINE_ the number of the line in the script being parsed
- PHP_VERSION the PHP version in use

- PHP_OS the operating system using PHP

User Defined Constant

constants.php *

```
<?
define("MYCONSTANT","This is a test of defining constants.");
echo MYCONSTANT;
?>
```

Pre-Defined Constants

constants2.php *

```
<?
echo "<br>This file is ".__FILE__;

echo "<br>This is line number ".__LINE__;

echo "<br>I am using ".PHP_VERSION;

echo "<br>This test is being run on ".PHP_OS;
?>
```

What is an Operator?

You have already used the assignment operation when assigning variables. The basic function of an operator is to something with the value of a variable.

Here are some of the main types of operators in PHP.
Assignment Operators (+=, -=, .=) as shown in the script below.

assignscript.php *

```
<HTML>
<HEAD>
<TITLE>Using Assignment Operators</TITLE>
</HEAD>
<BODY><H2>

<?
$origVar = 100;
echo "<P>Original value is $origVar</p>";

$origVar += 25;
echo "<P>Added a value, now it's $origVar</p>";

$origVar -= 12;
echo "<P>Subtracted a value, now it's $origVar</p>";

$origVar .= " chickens";
echo "<P>Final answer: $origVar</p>";
?>
</H2></BODY>
</HTML>
```

Arithmetic Operators

Of course they are to do math! As demonstrated in this script:

arithmeticscript.php

```
<HTML>
<HEAD>
<TITLE>Using Arithmetic Operators</TITLE>
</HEAD>
<BODY><H2>

<?
$a = 85;
$b = 24;
echo "<P>Original value of \$a is  $a and \$b is $b</p>";

$c = $a + $b;
echo "<P>Added \$a and \$b and got $c</p>";

$c = $a - $b;
echo "<P>Subtracted \$b from \$a and got $c</p>";

$c = $a * $b;
echo "<P>Multiplied \$a and \$b and got $c</p>";

$c = $a / $b;
echo "<P>Divided \$a by \$b and got $c</p>";

$c = $a % $b;
echo "<P>The modulus of \$a and \$b is $c</p>";
?>

</H2></BODY>
</HTML>
```

Comparison Operators

They do exactly that! But in order to do this we must use the IF ELSE statement. Compare one value to another as in this next script.

comparisonscript.php

```
<HTML>
<HEAD>
<TITLE>Using Comparison Operators</TITLE>
</HEAD>
<BODY><H1>

<?
$a = 21;
$b = 15;
echo "<P>Original value of \$a is  $a and \$b is $b</p>";


if ($a == $b) {
    echo "<P>TEST 1: \$a equals \$b</p>";
} else {
    echo "<P>TEST 1: \$a is not equal to \$b</p>";
}

if ($a != $b) {
    echo "<P>TEST 2: \$a is not equal to \$b</p>";
} else {
    echo "<P>TEST 2: \$a is equal to \$b</p>";
}

if ($a > $b) {
    echo "<P>TEST 3: \$a is greater than \$b</p>";
} else {
    echo "<P>TEST 3: \$a is not greater than \$b</p>";
}
```

```
30
31  if ($a < $b) {
32      echo "<P>TEST 4: \$a is less than \$b</p>";
33  } else {
34      echo "<P>TEST 4: \$a is not less than \$b</p>";
35  }
36
37  if ($a >= $b) {
38      echo "<P>TEST 5: \$a is greater than or equal to \$b</p>";
39  } else {
40      echo "<P>TEST 5: \$a is not greater than or equal to \$b</p>";
41  }
42
43  if ($a <= $b) {
44      echo "<P>TEST 6: \$a is less than or equal to \$b</p>";
45  } else {
46      echo "<P>TEST 6: \$a is not less than or equal to \$b</p>";
47  }
48  ?>
49  </H1></BODY>
50  </HTML>
51
```

Logical Operators

Logical operator are the And (&&) and Or (||). Again we will need to use the IF ELSE statements to demonstrate as seen in this next script.

logicalscript.php *

```
1   <HTML>
2   <HEAD>
3   <TITLE>Using Logical Operators</TITLE>
4   </HEAD>
5   <BODY><H1>
6
7   <?
8   $degrees = "95";
9   $hot = "yes";
10
11  if (($degrees >  100) || ($hot == "yes")) {
12      echo "<P>TEST 1: It's <strong>really</strong> hot!</p>";
13  } else {
14      echo "<P>TEST 1:It's bearable.</p>";
15  }
16
17  if (($degrees >  100) && ($hot == "yes")) {
18      echo "<P>TEST 2: It's <strong>really</strong> hot!</p>";
19  } else {
20      echo "<P> TEST 2: It's bearable.</p>";
21  }
22
23  ?>
24  </H1></BODY>
25  </HTML>
26
27
```

Lab 2

Create the following scripts and have me sign off on the

- **Variables and Types – printvarscript.php**
- **Constants – constants.php**
- **Pre-Defined Constants – constants2.php**
- **Assignment Operators – assignscript.php**

- **Arithmetic Operators – arithmeticscript.php**
- **Comparison Operators – comparescript.php**
- **Logical Operators – logicalscript.php**