SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : 19SB602 FULL STACK DEVELOPMENT FOR NEXT GENERATION IOT

III YEAR / VI SEMESTER

Unit III- **CORE PHP WITH MODEL–VIEW–CONTROLLER**

Topic : Data Types, Strings, Operators

Data Types

String

Integer

Float (floating point numbers - also called double)

Boolean

Array

Object
NULL
Resource

Get the Type

To get the data type of a variable, use the var_dump() function.

Example

The var_dump() function returns the data type and the value:

```
$x = 5;
var_dump($x);
```

Example

See what var_dump() returns for other data types:

```
var_dump(5);
var_dump("John");
var_dump(3.14);
var_dump(true);
var_dump([2, 3, 56]);
var_dump(NULL);
```

Assign String to a Variable

Assigning a string to a variable is done with the variable name followed by an equal sign and the string:

Example

```
$x = "John";
echo $x;
```

Assign Multiple Values

You can assign the same value to multiple variables in one line:

Example

All three variables get the value "Fruit":

```
$x = $y = $z = "Fruit";
```

PHP Variables Scope

In PHP, variables can be declared anywhere in the script.

The scope of a variable is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

local
global
static

Global and Local Scope

A variable declared outside a function has a GLOBAL SCOPE and can only be accessed outside a function:

Example

Variable with global scope:

```
$x = 5; // global scope
function myTest() {
  // using x inside this function will generate an error
  echo "<p>Variable x inside function is: $x</p>";
}
myTest();

echo "<p>Variable x outside function is: $x</p>";
```

PHP The global Keyword

The global keyword is used to access a global variable from within a function.

To do this, use the global keyword before the variables (inside the function):

Example

```
$x = 5;
$y = 10;

function myTest() {
  global $x, $y;
  $y = $x + $y;
}
```

PHP The Static Keyword

Normally, when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job.

To do this, use the static keyword when you first declare the variable:

Example

```
function myTest() {
  static $x = 0;
  echo $x;
  $x++;
}
myTest();
myTest();
myTest();
```

The PHP echo Statement

The echo statement can be used with or without parentheses: echo or echo().

Display Text

The following example shows how to output text with the echo command (notice that the text can contain HTML markup):

Example

```
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
```

PHP Strings

A string is a sequence of characters, like "Hello world!".

Strings

Strings in PHP are surrounded by either double quotation marks, or single quotation marks.

Example

```
echo "Hello";
echo 'Hello';
```

Double or Single Quotes?

You can use double or single quotes, but you should be aware of the differences between the two.

Double quoted strings perform action on special characters.

E.g. when there is a variable in the string, it returns the value of the variable:

Example

Double quoted string literals perform operations for special characters:

```
$x = "John";
echo "Hello $x";
```

Single quoted strings does not perform such actions, it returns the string like it was written, with the variable name:

Example

Single quoted string literals returns the string as it is:

```
$x = "John";
echo 'Hello $x';
```

String Length

The PHP strlen() function returns the length of a string.

Example

Return the length of the string "Hello world!":

```
echo strlen("Hello world!");
```

Word Count

The PHP str_word_count() function counts the number of words in a string.

Example

Count the number of word in the string "Hello world!":

```
echo str_word_count("Hello world!");
```

Search For Text Within a String

The PHP strpos() function searches for a specific text within a string.

If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.

Example

Search for the text "world" in the string "Hello world!":

```
echo strpos("Hello world!", "world");
```

PHP - Modify Strings

PHP has a set of built-in functions that you can use to modify strings.

Upper Case

The strtoupper() function returns the string in upper case

Example

```
$x = "Hello World!";
echo strtoupper($x);
```

Lower Case

The strtolower() function returns the string in lower case

Example

```
$x = "Hello World!";
echo strtolower($x);
```

Replace String

The PHP str_replace() function replaces some characters with some other characters in a string.

Example

Replace the text "World" with "Dolly":

```
$x = "Hello World!";
echo str_replace("World", "Dolly", $x);
```

Reverse a String

The PHP strrev() function reverses a string.

Example
Reverse the string "Hello World!":

```
$x = "Hello World!";
echo strrev($x);
```

Remove Whitespace

Whitespace is the space before and/or after the actual text, and very often you want to remove this space.

Example

The trim() removes any whitespace from the beginning or the end:

```
$x = " Hello World! ";
echo trim($x);
```

String Concatenation

To concatenate, or combine, two strings you can use the . operator:

Example

```
$x = "Hello";
$y = "World";
$z = $x . $y;
echo $z;
```

PHP - Slicing Strings

Slicing

You can return a range of characters by using the substr() function.

Specify the start index and the number of characters you want to return.

Example

Start the slice at index 6 and end the slice 5 positions later:

```
$x = "Hello World!";
echo substr($x, 6, 5);
```

PHP - Escape Characters

Escape Character

To insert characters that are illegal in a string, use an escape character.

An escape character is a backslash \ followed by the character you want to insert.

An example of an illegal character is a double quote inside a string that is surrounded by double quotes:

Example

```
$x = "We are the so-called "Vikings" from the north.";
```

PHP Operators

Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

Arithmetic operators
Assignment operators
Comparison operators
Increment/Decrement operators
Logical operators
String operators
Array operators
Conditional assignment operators

PHP Arithmetic Operators

The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power

PHP Comparison Operators

The PHP comparison operators are used to compare two values (number or string)

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y
<=>	Spaceship	$x <=> $y	Returns an integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y. Introduced in PHP 7.

PHP Increment / Decrement Operators

The PHP increment operators are used to increment a variable's value.

The PHP decrement operators are used to decrement a variable's value.

Operator	Same as...	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

PHP Logical Operators

The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

PHP String Operators

PHP has two operators that are specially designed for strings.

Operator	Name	Example	Result
.	Concatenation	$txt1 . $txt2	Concatenation of $txt1 and $txt2
.=	Concatenation assignment	$txt1 .= $txt2	Appends $txt2 to $txt1

PHP Array Operators

The PHP array operators are used to compare arrays.

Operator	Name	Example	Result
+	Union	$x + $y	Union of $x and $y
==	Equality	$x == $y	Returns true if $x and $y have the same key/value pairs
===	Identity	$x === $y	Returns true if $x and $y have the same key/value pairs in the same order and of the same types
!=	Inequality	$x != $y	Returns true if $x is not equal to $y
<>	Inequality	$x <> $y	Returns true if $x is not equal to $y
!==	Non-identity	$x !== $y	Returns true if $x is not identical to $y

PHP Conditional Assignment Operators

The PHP conditional assignment operators are used to set a value depending on conditions

Operator	Name	Example	Result
?:	Ternary	$x = *expr1* ? *expr2* : *expr3*	Returns the value of $x. The value of $x is *expr2* if *expr1* = TRUE. The value of $x is *expr3* if *expr1* = FALSE
??	Null coalescing	$x = *expr1* ?? *expr2*	Returns the value of $x. The value of $x is *expr1* if *expr1* exists, and is not NULL. If *expr1* does not exist, or is NULL, the value of $x is *expr2*. Introduced in PHP 7

Any Query????

Thank you……