Tutorial 09 – PHP Basics

SOEN 287

Mohamed Eldesouki

soen287@eldesouki.ca

Tutorial Sections UC & SB

Introduction to PHP

- **PHP** is an acronym for "PHP: Hypertext Preprocessor"
- **PHP** is a widely-used, open source scripting language
- **PHP** scripts are executed on the server.
- **PHP** is used to develop WordPress and Facebook.

Introduction to PHP

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, NGiNX, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

Introduction to PHP

PHP can create, open, read, write, delete, and close files on the server

PHP can collect form data

PHP can send and receive cookies

PHP can add, delete, modify data in your database

PHP can generate dynamic page content

Introduction to PHP

- To write PHP you need to create a PHP file.
- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

Installing PHP

Two main options

1. Find a web host with PHP and MySQL support.

2. Install a web server on your own PC, and then install PHP and MySQL,

(a) Manually installing PHP, MySQL and Apache,

(b) Pre-configured Packages such as XAMPP and WAMP,

(c) Virtual Machines and Vagrant,

(d) or finally Docker

PHP Language (Basic)

A PHP script can be placed anywhere in the document.

A PHP script starts with <?php and ends with ?>

```
<?php
// PHP code goes here
?>
```

Comments in PHP

is a line that is not executed as a part of the program.

Purpose of Comment:

- Let others understand your code
- Remind yourself of what you did
- Leave out parts of the code

```
<?php

// This is a single-line comment

# This is also a single-line comment

/*
This is a multiple-lines comment block
that spans over multiple lines
*/
?>
```

PHP Variables

- In PHP, a variable starts with the $ sign, followed by the name of the variable

```php
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

- When you assign a text value to a variable, put quotes around the value.

~~- When you assign a text value to a variable, put quotes around the value.~~

- PHP has no command for declaring a variable. It is created the moment you first assign a value to it.

- Variable names are case-sensitive ($name and $NAME are two different variables)

PHP Variables

Use ECHO, PRINT to out data to screen

ECHO	PRINT
no return value	return value of 1
can take multiple parameters	take one argument
faster	

PHP commands are not case-sensitive

```
<?php
$txt = "PHP";
echo "I love $txt!";
ECHO "I love " . $txt . "!";
?>
```

PHP is a Loosely Typed Language

PHP Variables

Scope: is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

- Local -> within a function
- Global -> outside a function
- Static -> local variable that is not deleted

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

The static keyword is used to declare the static variable

PHP Data Types

PHP supports the following data types:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource (the storing of a reference to functions and resources external to PHP such as a DB call)

PHP Data Types

- The var_dump() function dumps information about one or more variables. The information holds type and value of the variable(s).
- Use getType() to detect types of variables.
- Use isSet() to detect unbound variables; use empty to determine whether an object is null or empty.

PHP Strings

strlen()

str_word_count()

strrev()

strpos()

str_replace()

PHP Numbers

- Numbers in PHP can be **Integers**, **Floats**, and **Number Strings**
- Integers can be specified in three formats:
  - decimal (10-based),
  - hexadecimal (16-based - prefixed with 0x) or
  - octal (8-based - prefixed with 0)
- Use is_nan() to check if a value is not a number,
- Use is_int() to check if the type of a variable is integer,
- Use is_float() to check if the type of a variable is float
- Use is_numeric() to check if the type is numeric

PHP if...else...elseif Statements

```php
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day!";
}
?>
```

```php
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

PHP if...else...elseif Statements

```php
<?php
$t = date("H");

if ($t < "10") {
  echo "Have a good morning!";
} elseif ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

PHP switch Statement

```php
<?php
$favcolor = "red";

switch ($favcolor) {
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

PHP Loops

In PHP, we have the following loop types:

- `while` - loops through a block of code as long as the specified condition is true
- `do...while` - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- `for` - loops through a block of code a specified number of times
- `foreach` - loops through a block of code for each element in an array

The PHP Loops: while vs. do while

```php
<?php
$x = 1;

while($x <= 5) {
  echo "The number is: $x <br>";
  $x++;
}
?>
```

```php
<?php
$x = 1;

do {
  echo "The number is: $x <br>";
  $x++;
} while ($x <= 5);
?>
```

The PHP Loops: for vs. foreach

```php
<?php
for ($x = 0; $x <= 10; $x++) {
  echo "The number is: $x <br>";
}
?>
```

```php
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
  echo "$value <br>";
}
?>
```

The PHP Loops: for vs. foreach

```php
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $val) {
  echo "$x = $val<br>";
}
?>
```

PHP Functions

```php
<?php
function writeMsg() {
  echo "Hello world!";
}

writeMsg(); // call the function
?>
```

```php
<?php
function familyName($fname, $year) {
  echo "$fname Refsnes. Born in $year <br>";
}

familyName("Hege", "1975");
familyName("Stale", "1978");
familyName("Kai Jim", "1983");
?>
```

References

(1)https://www.sitepoint.com/docker-php-development-environment/

(2)https://www.w3schools.com/php/default.asp

(3)Other Concordia tutorial slides for the this course [SOEN287]