



UTM
UNIVERSITI TEKNOLOGI MALAYSIA



**Online
Learning**

PHP - Hypertext Preprocessor (Part 1)

Sarina Sulaiman
Faculty of Computing

Introduction

- PHP is a **powerful server-side scripting language** for creating dynamic and interactive websites.
- PHP is the widely-used, **free**, and efficient alternative to competitors such as Microsoft's ASP.
- PHP is perfectly suited for Web development and can be **embedded** directly into the HTML code.
- The PHP syntax is very similar to Perl and C.
- PHP is often used together with **Apache** (web server) on various operating systems.

What is PHP?

- PHP stands for PHP: **Hypertext Preprocessor**
- PHP is a server-side scripting language, like ASP
- PHP scripts are **executed on the server** even though combined with an HTML code
- PHP **supports many databases** (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP is **an open source software** (OSS)
- PHP is **free to download** and use

What is a PHP File?



PHP files may contain text, HTML tags and scripts



PHP files are returned to the browser as plain HTML



PHP files have a file extension of ".php", ".php3", or ".phtml"

What is MySQL?

MySQL is a database server



MySQL is ideal for both small and large applications



MySQL supports standard SQL



MySQL compiles on a number of platforms



MySQL is free to download and use

PHP + MySQL

- PHP combined with MySQL are **cross-platform** (means that you can develop in Windows and serve on a Unix platform)

Why PHP?

- PHP runs on **different platforms** (Windows, Linux, Unix, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP is FREE to download from the official PHP resource: www.php.net
- PHP is **easy to learn** and **runs efficiently** on the server side

Where to Start?

Install an Apache server on a Windows or Linux machine

Install PHP on a Windows or Linux machine

Install MySQL on a Windows or Linux machine

Install PHP triad

- All in one
 - LAMP
 - Linux Apache MySQL PHP

First Example – Hello World!

■ Activity 01

■ NOTES

- You cannot view the PHP source code by selecting "View source" in the browser
- you will only see the output from the PHP file, which is plain HTML.
- This is because the **scripts are executed on the server** before the result is sent back to the browser.

Activity 01

```
<html>

<head>
<title>PHP First App - Hello World</title>

<style type="text/css">
<!--
.myStyle1 {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 14px;
    font-weight: normal;
    color: #0000FF;
    background-color: #FFFF00;
}
-->
</style>

</head>
<body>

<?php
    echo '<b class="myStyle1">Hello World</b>';
?>

</body>
</html>
```

Output

Hello World



```
<?php
    print '<b class="myStyle1">Hello World</b>';
?>
```

PHP Syntax

- A PHP scripting block always starts with **<?php** and ends with **?>**
- A PHP scripting block can be **placed anywhere** in the HTML document
- On servers with shorthand support enabled you can start a scripting block with **<? and end with ?>**
- However, for maximum compatibility, it is recommend that you use the standard form (**<?php**) rather than the shorthand form

PHP Syntax

- A PHP file normally contains **HTML tags**, just like an HTML file, and some **PHP scripting code**.
- Each code line in PHP must **end with a semicolon**.
- The semicolon is a **separator** and is used to distinguish **one set of instructions** from another.
- There are two basic statements to output text with PHP: **echo and print**.
- In **activity 01**, we have used the echo statement to output the text "Hello World".

PHP Syntax

- Comments in PHP
 - In PHP, we use `//` to make a **single-line comment** or
 - `/*` and `*/` to make a **large comment block**.

PHP Variables

- All variables in PHP start with a **\$ sign symbol**.
- Variables may contain strings, numbers, or arrays (untyped).
- **Activity 02: Example 01**
- Variable concatenation – using the (.) operator
 - **All converted to string**
 - Str . Str = str
 - Num . Num = str
- **Activity 02: Example 02**

Activity 02: Example 01 - Using the \$ for var

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>PHP Variable</title>
</head>

<?php
$txt="Hello World";
echo $txt;
?>

<body>
</body>
</html>
```

Output

Hello World

Activity 02: Example 02 – Variable concatenation

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Variable concatenation</title>
</head>
<body>

<?php
$txt1 = "Hello World";
$txt2 = "1234";
echo $txt1 . ' ' . $txt2."<br />";
$num1 = 2;
$num2 = 3;
echo $num1 + $num2;
echo '<br />';
echo $num1.$num2;
?>

</body>
</html>
```

Output

Hello World 1234
5
23

Variable Naming Rules

- Same as C, C++ or Java
- A variable name must **start with a letter or an underscore "_"**
- A variable name can only contain **alpha-numeric characters and underscores (a-Z, 0-9, and _)**
- A variable name should **not contain spaces**.
- If a variable name should be more than one word, it should be separated with **underscore (\$my_string)**,
- or with **capitalization (\$myString)**

HTML out

- concat and no concat - source
- A tag - source
- IMG tag - source
- TABLE tag - source

PHP operator, conditional

- Same as C or C++
- PHP If...Else Statements – **Activity 04**
 - Same as C or C++
- Switch statement
 - Same as C or C++

PHP operator, conditional

```
<html>
<body>

<?php
switch ($x)
{
case 1:
    echo "Number 1";
    break;
case 2:
    echo "Number 2";
    break;
case 3:
    echo "Number 3";
    break;
default:
    echo "No number between 1 and 3";
}
?>

</body>
</html>
```

Activity 04: PHP operator, conditional

- [Example 01](#) - if else - [source](#)
- [Example 02](#) - if else (compound statement) - [source](#)
- [Example 03](#) - if else if - [source](#)

PHP array

- **Numeric Arrays**
- `$names = array("Peter","Quagmire","Joe");`
- Assign manually:
 - `$names[0] = "Peter";`
 - `$names[1] = "Quagmire";`
 - `$names[2] = "Joe";`
- Assign directly:
 - `$family = array('Fred', 'Wilma');`

PHP array

- Adding Values to the **End of an Array**
 - `$family[] = 'Pebbles';` // `$family[2]` is 'Pebbles'
- Getting the **Size of an Array**
 - `$size = count($family);` // `$size` is 3

PHP array

- **Traversing Arrays**
 - `$family = array('Fred', 'Wilma');`
 - `foreach($family as $name) {`
 - `for($i = 0; $i < count($family); $i++) {`

Activity 05: PHP array

- [Example 01](#) - numerical array - [source](#)
- [Multidimensional array](#) - [source](#)
- TABLE tag re-visit - [array in table](#) - [source](#)

PHP Looping – Activity 06

- Same as C or C++
- **while** - loops through a block of code if and as long as a specified condition is true
- **do...while** - loops through a block of code once, and then repeats the loop as long as a special 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

Activity 06: PHP looping

- [Example 01](#) - while loop html through echo - [source](#)
- [Example 02](#) - while loop - full html - [source](#)
- [Example 03](#) - foreach - [source](#)



UTM
UNIVERSITI TEKNOLOGI MALAYSIA



**Online
Learning**

<http://gmm.fsksm.utm.my/~rosely/scv1223/php/>

THANK YOU