```
<?
   PHP Code In Here
?>

<?php
   PHP Code In Here
php?>

<script language="php">
   PHP Code In Here
</script>
```

04 PHP I

PHP I

- Introduction to PHP
- PHP Syntax
- Operators
- Workshop

Introduction to PHP

- What is PHP?
- Why PHP?
- Some of PHP's Strengths
- PHP Usage Statistic
- How PHP Works?
- What Does PHP Look Like?
- Declaring PHP
- A Simple Example and Its Process

Introduction to PHP

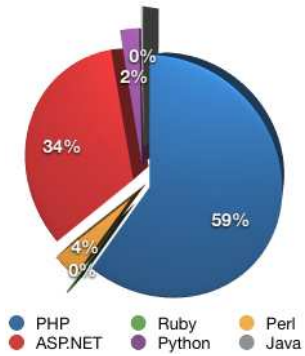
- What is PHP?
  - PHP originally stood for Personal Home Page.
  - Now, PHP stands for PHP Hypertext Preprocessor.
  - It was created in 1995 by **Rasmus Lerdorf.**
  - PHP is a server-side scripting language, like ASP.
  - PHP scripts are executed on the server.
  - PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.).
  - PHP is an open source software (OSS).
  - PHP is free to download and use.

- 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: http://www.php.net
  - PHP is easy to learn and runs efficient on the server side.
  - PHP is a simple scripting language embedded into a html page
  - PHP can be used to insert, search and retrieve information from a database.
  - PHP is very similar to C language and especially for web development.

- Some of PHP's Strengths
  - PHP has many strengths over its competitors e.g., Perl, Microsoft Active Server Pages (ASP), and Java Server Pages (JSP) including:
    - High Performance
    - Interface to many different database systems
    - Built-in library for many common Web tasks
    - Low cost
    - Ease of learning and use
    - Portability
    - Available of source code

- PHP Usage Statistic

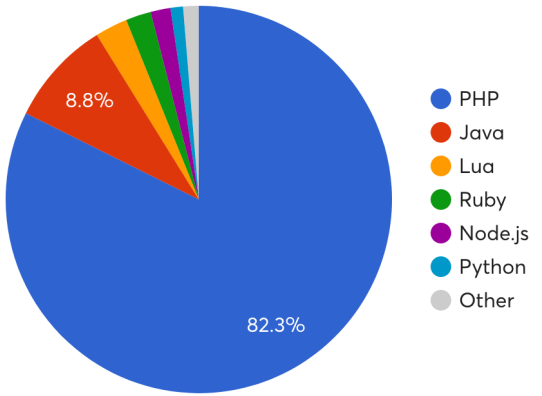
PHP	3998425	59%
ASP.NET	2294166	34%
Perl	259931	4%
Python	159475	2%
Java	18065	0%
Ruby	16539	0%



The above chart shows the breakdown of the 6 major, identifiable languages from 6.7 million domains where the language could be determined.

Source: http://phpadvent.org/2010/usage-statistics-by-ilia-alshanetsky

- PHP Usage Statistic (cont.)



Wappalyzer stats indicate PHP has a 82% marketshare in 2018.

Source: https://react-etc.net/entry/wappalyzer-stats-indicate-php-has-a-82-marketshare-in-2018

How PHP Works?

- PHP scripts are **embedded** in HTML files accessible via a web server (file extension .php)
- A browser is used to access an HTML file **with an embedded script**
- The **web server passes the HTML file** to the PHP pre-processor
- The PHP pre-processor…
  - processes the embedded PHP scripts
  - any resulting output is **substituted for the text of the script**
  - the resulting HTML data is sent **back to the web server**
- The **web server sends the HTML** data to the browser

What Does PHP Look Like?

- PHP is a programming language like any other it allows you to create scripts which **manipulate data** and then **output the results** into the body of an HTML file.
- A PHP element looks like…

```
<?php
   … lines of php script…
?>
```

- Rules…
  - PHP can be embedded **anywhere** in an HTML document.
  - **Multiple** PHP elements can be embedded throughout a document.
  - The HTML file should have the extension **.php**

Declaring PHP

- PHP scripts are always enclosed in between two PHP tags.
- This tells your server to parse the information between them as PHP.
- The four different styles are as follows:

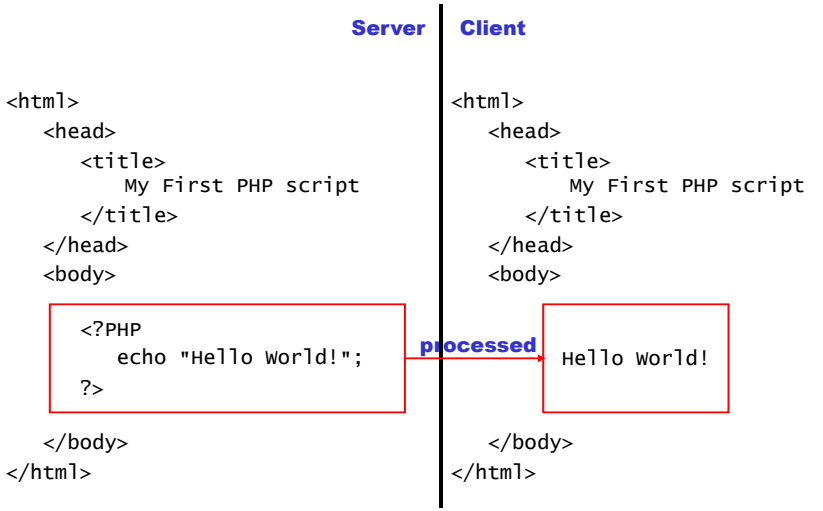
```
XML Style
<?php
   PHP Code In Here
?>
```

```
SCRIPT Style
<script language="php">
   PHP Code In Here
</script>
```

```
Short style
<?
   PHP Code In Here
?>
```

```
ASP Style
<%
   PHP Code In Here
%>
```


A Simple Example and Its Process



PHP Syntax

- Basic PHP Syntax
- Variables in PHP
- ECHO Statement
- Comments in PHP
- The Order Form
- Adding Dynamic Content
- Accessing Form Variables
- Variable Types
- Constants
- Variable Scope

PHP Syntax

- Basic PHP Syntax
  - A PHP scripting block always starts with <?php and ends with ?>... As expected to be the most appropriate style!
  - A PHP scripting block can be placed anywhere in the document.
  - 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.

PHP Syntax

- Variables in PHP
  - All variables in PHP start with a $ sign symbol.
  - Variables may contain strings, numbers, or arrays.
  - Variables are **case sensitive**
    - $Name is not the same as $name
    - For example:
      - $Name = "Sombat";
      - $name = "Somsri";
  - To concatenate two or more variables together, use the dot (.) operator.

```
<html>
  <body>
  <?php
    $txt="Hello World";
    echo $txt;
  ?>
  </body>
</html>
```

```
<html>
  <body>
  <?php
    $txt1="Hello World";
    $txt2="1234";
    echo $txt1." ".$txt2 ;
  ?>
  </body>
</html>
```


PHP Syntax

- ECHO Statements
  - Single (') or Double (") Quotation Marks?
    - Single quotation mark uses to:
      - Display exactly what was inside
      - For example

```
» $num = 1;
» echo '$num';      >>> $num
```

    - Double quotation mark uses to:
      - Calculate it first if it is a variable
      - For example:

```
» $num = 1;
» echo "$num";    >>> 1
» echo $num;      >>> 1
```

    - **SUGGESTION!**
      - Sometime a single quotation mark may cause a problem!
      - Better to use Double Quotation

PHP Syntax

- Comments in PHP
  - Use // to make a single-line comment or /* and */ to make a large comment block.

```
<html>
  <body>
  <?php
    //This is a comment
    /* This is a
    comment block
    */
  ?>
  </body>
</html>
```

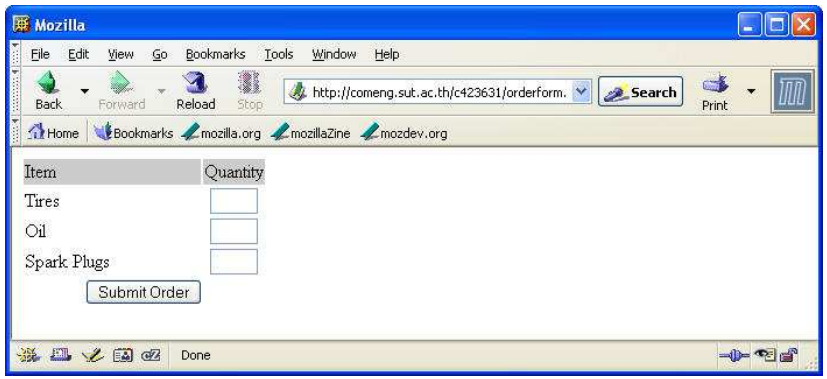

PHP Syntax

- The Order Form

```
<form action="processorder.php" method="post">
<table border="0">
<tr bgcolor="#cccccc">
  <td width="150">Item</td> <td width="150">Quantity</td>
</tr>
<tr>
  <td>Tires</td>
  <td align="center"><input type="text" name="tireqty" size="3" maxlength="3"></td>
</tr>
<tr>
  <td>Oil</td>
  <td align="center"><input type="text" name="oilqty" size="3" maxlength="3"></td>
</tr>
<tr>
  <td>Spark Plugs</td>
  <td align="center"><input type="text" name="sparkqty" size="3" maxlength="3"></td>
</tr>
<tr>
  <td colspan="2" align="center"><input type="submit" value="Submit Order"></td>
</tr>
</table>
</form>
```


PHP Syntax

- The Order Form (cont.)



PHP Syntax

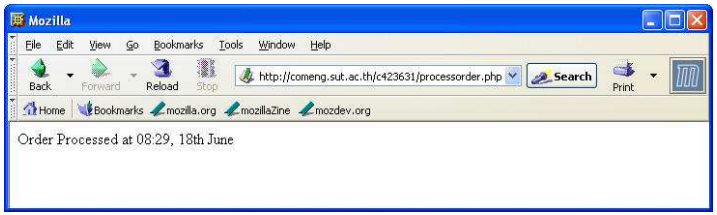
- Adding Dynamic Content
  - Using a server-side scripting language to provide dynamic content to a site's user

```
<?php
  echo "<p>Order Processed at";
  echo date("H:i, jS F");
  echo "</p>";
?>
```

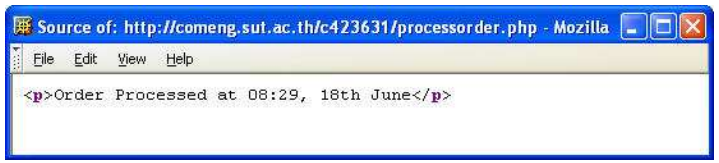
  - Calling date() function
    - date() function expects the argument to be a format string.
      - *H* is the hour in a 24-hour format,
      - *i* is the minutes which with a leading zero where required,
      - *j* is the day of the month without the leading zero,
      - *S* represents the ordinary suffix (st, nd, rd and th), and
      - *F* is the full name of the month.

Adding Dynamic Content (cont.)

- Output



- Preview source code from the browser (View > Page Source)



Accessing Form Variables

```
… <input type="text" name="tireqty" size="3" maxlength="3"> …
… <input type="text" name="oilqty" size="3" maxlength="3"> …
… <input type="text" name="sparkqty" size="3" maxlength="3"> …
```

- To access the content of the field "tireqty" we can simply use:
  - $tireqty // short style
  - $_POST["tireqty"] // medium style
  - $HTTP_POST_VARS["tireqty"] // long style

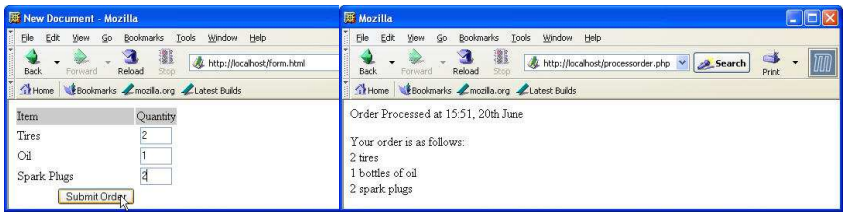
  - **Short Style:** is convenient, but requires the register globals configuration setting to be on. (file: php.ini)
  - **Medium Style:** is fairly convenient, but only came into existence with PHP 4.1.0, so it won't work on older installations.
  - **Long Style**: is the most verbose, but it is the only style that is guaranteed at present to work on every server, regardless of the configuration. Note, however, that it is deprecated and is therefore likely to be removed in the long term.

Accessing Form Variables (cont.)

- So, the complete PHP code to display the value of each variable from the html form:

```
<?php
  echo "<p>Order Processed at ";
  echo date("H:i, jS F");
  echo "</p>";
  echo "Your order is as follows:<br>";
  echo "$tireqty tires<br>";
  echo "$oilqty bottles of oil<br>";
  echo "$sparkqty spark plugs<br>";
?>
```

```
// must be declared if register global
// is turn off.
$tireqty = $_POST["tireqty"];
$oilqty = $_POST["oilqty"];
$sparkqty = $_POST["sparkqty"];
```



Variable Types

- PHP's Data Types
  - The kind of data that is stored in a variable
    - Integer: Used for whole numbers
    - Double: Used for real numbers
    - String: Used for strings of characters
    - Boolean: Used for true or false values
    - Array: Used to store multiple data items of the same type
    - Object: Used for storing instances of classes
- Type Casting
  - Enable to convert a defined type to be a different type
    - $totalqty = 0;
    - $totalamount = (double)$totalqty;
- Variable Variables
  - Enable to change the name of a variable dynamically
    - $varname = 'tireqty';
    - $tireqty;
    - $$varname = 5; is equivalent to $tireqty = 5

PHP Syntax

- Constants
  - A constant stores a value such as a variable, but its value is set once and then cannot be changed elsewhere in the script.
    - define("TIREPRICE", 100);
    - define("OILPRICE", 10);
    - define("SPARKPRICE", 4);

```
<?php
  define("TIREPRICE", 100);
  define("OILPRICE", 10);
  define("SPARKPRICE", 4);
  echo "<p>Order Processed at ";
  echo date("H:i, jS F");
  echo "</p>";
  echo "Your order is as follows:<br/>";
  echo "$tireqty tires "."(@$".TIREPRICE.")<br/>";
  echo "$oilqty bottles of oil "."(@$".OILPRICE.")<br/>";
  echo "$sparkqty spark plugs "."(@$".SPARKPRICE.")<p/>";
?>
```


PHP Syntax

- Variable Scope
  - The four types of scope in PHP are as follows:
    - Built-in superglobal variables are visible everywhere within a script.
    - Global variables declared in a script are visible throughout that script, but *not inside functions*.
    - Variables used inside functions are local to the function.
    - Variables used inside functions that are declared as global refer to the global variable of the same name.

The complete list of superglobals is as follows:

- $GLOBALS an array of all global variables
- $_SERVER, an array of server environment variables
- $_GET an array of variables passed to the script via the GET method
- $_POST an array of variables passed to the script via the POST method
- $_COOKIE an array of cookie variables
- $_FILES an array of variables related to file uploads
- $_ENV an array of environment variables
- $_REQUEST an array of all user input variables
- $_SESSION an array of session variables

Operators

- Arithmetic Operators
- Assignment Operators
- String Operators
- Increment and Decrement Operators
- Comparison Operators
- Logical Operators
- Error Control Operators
- Execution Operators
- Array Operators
- Type Operators
- Conditional Operators
- Operator Precedence

Operators

- Arithmetic Operators (+, -, *, /, %)
  - $sum = $num1 + $num2;
- Assignment Operators (=, +=, -=, *=, /=, %=)
  - $total += $num;
- String Operators (.)

```
<?php
  $msg1 = "Hello";
  $name = "Somchai Jaidee.";
  $age = 25;
  echo $msg1.", ".$name."<br>"."Your age is ".$age.".";
?>
```

```
Hello, Somchai Jaidee.
Your age is 25.
```

Operators

- Increment and Decrement Operators (++, --)
  - ++$total;
  - $count--;
- Comparison Operators (==, ===, !=, <>, !==, <, >, <=, >=)
  - $a != $b
- Logical Operators (&&, and, ||, or, xor, !)
  - (($a > $b) && ($a > $c))
- Error Control Operators (@)
  - $result = @(25/0);

Operators

- Execution Operators (`)
  - $output = `dir D:`;
  - echo '<pre>'.$output.'</pre>';
- Array Operators (+, ==, ===, !=, <>, !==)
  - $a === $b;

  (จะเป็นจริงก็ต่อเมื่อ คีย์/ค่า (key/value) ของทั้ง 2 คู่เหมือนกันและเป็นชนิดเดียวกัน)

  ($a + $b คือการ Union of $a and $b)
- Conditional Operators
  - $x = 5;
  - $y = 10;
  - $min = ($x < $y) ? $x : $y;

Operators

- Operator Precedence

ความสัมพันธ์	ตัวดำเนินการ	ลำดับความสำคัญ
ไม่มี	new	สูงสุด
ซ้ายไปขวา	[	
ไม่มี	++ --	
ไม่มี	~ - (int) (float) (string) (array) (object) @	
ไม่มี	instanceof	
ขวาไปซ้าย	!	
ซ้ายไปขวา	* / %	
ซ้ายไปขวา	+ - .	
ซ้ายไปขวา	<< >>	
ไม่มี	< <= > >=	
ไม่มี	== != === !===	
ซ้ายไปขวา	&	
ซ้ายไปขวา	^	
ซ้ายไปขวา	|	
ซ้ายไปขวา	&&	
ซ้ายไปขวา	||	
ซ้ายไปขวา	? :	
ขวาไปซ้าย	= += -= *= /= %= .= &= \= ^= <<= >>=	
ซ้ายไปขวา	and	
ซ้ายไปขวา	xor	
ซ้ายไปขวา	Or	
ซ้ายไปขวา	,	ต่ำสุด

Workshop

- PHP I
  - Laboratory: Page 80 (1-2)