|  |
| --- |
| **โครงสร้างของภาษา PHP** |
| **PHP คืออะไร** |
|      ในช่วงแรกภาษาที่นิยมใช้งานบนระบบเครือข่าย คือ ภาษา HTML (Hypertext Markup Language) แต่ภาษา HTML มีลักษณะเป็น Static คือ ภาษาที่มีลักษณะของข้อมูลคงที่ ซึ่งไม่เพียงพอต่อความต้องการในปัจจุบันที่นิยมใช้ระบบเครือข่าย Internet เป็นศูนย์กลางในการติดต่อระหว่างกัน ทำให้ต้องการใช้เว็บไซต์ที่มีลักษณะเป็นแบบ Dynamic คือ เว็บไซต์ที่ข้อมูลสามารถเปลี่ยนแปลงได้โดยอัตโนมัติตามเงื่อนไขต่าง ๆ ที่ผู้เขียนเว็บไซต์เป็นผู้กำหนด และการควบคุมการทำงานเหล่านี้จะกระทำโดยโปรแกรมภาษาสคริปต์ เช่น ภาษา PHP ซึ่งเป็นภาษาหนึ่งที่ได้รับความนิยมเป็นอย่างมากในปัจจุบัน     PHP ถูกสร้างขึ้นในปี ค.ศ.1994 โดย [Rasmus Lerdorf](http://lerdorf.com/bio.php) ต่อมามีผู้ให้ความสนใจเป็นจำนวนมาก จึงได้ออกเป็นแพ็คเกจ "Personal Home Page" ซึ่งเป็นที่มาของ PHP โดยภาษา PHP เป็นแบบ Server Side Script และเป็น Open Source ที่ผู้ใช้ทั่วไปสามารถดาวน์โหลด Source Code และโปรแกรมไปใช้ฟรี ได้ที่ [http://www.php.net](http://www.php.net/)      พอกลางปี ค.ศ.1995 เขาก็ได้พัฒนาตัวแปลภาษา PHP ขึ้นมาใหม่ โดยใช้ชื่อว่า PHP/FI เวอร์ชั่น 2 ซึ่งได้เพิ่มความสามารถในการรับข้อมูลที่ส่งมาจากฟอร์มของ HTML (จึงมีชื่อว่า FI หรือ Form Interpreter) นอกจากนั้นยังเพิ่มความสามารถในการติดต่อกับฐานข้อมูลอีกด้วย จึงทำให้ผู้คนเริ่มหันมาสนใจ PHP กันมากขึ้น      ในปี 1997 มีผู้ร่วมพัฒนา PHP เพิ่มอีก 2 คน คือ [Zeev Suraski](http://www.zend.com/zend/founders.php%22%20%5Ct%20%22_blank) และ [Andi Gutmans](http://en.wikipedia.org/wiki/Andi_Gutmans%22%20%5Ct%20%22_blank) (กลุ่มที่เรียกตัวเองว่า Zend ซึ่งย่อมาจาก **Ze**ev และ A**nd**i ) โดยได้แก้ไขข้อบกพร่องต่างๆ และเพิ่มเติมเครื่องมือให้มากขึ้น |
| **โครงสร้างของภาษา PHP** |
|      ภาษา PHP มีลักษณะเป็น embedded script หมายความว่าเราสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจร่วมกับคำสั่ง(Tag) ของ HTML ได้ และสร้างไฟล์ที่มีนามสกุลเป็น .php, .php3 หรือ .php4 ซึ่งไวยากรณ์ที่ใช้ใน PHP เป็นการนำรูปแบบของภาษาต่างๆ มารวมกันได้แก่ C, Perl และ Java ทำให้ผู้ใช้ที่มีพื้นฐานของภาษาเหล่านี้อยู่แล้วสามารถศึกษา และใช้งานภาษานี้ได้ไม่ยาก  |
| ตัวอย่างที่ 1 |
|

|  |  |
| --- | --- |
| 12345678910 | <html> <head> <title>Example 1 </title> </head> <body>**<?    echo"Hi, I'm a PHP script!"; ?>** </body> </html> |

 |
|      จากตัวอย่าง บรรทัดที่ 6 - 8 เป็นส่วนของสคริปต์ PHP ซึ่งเริ่มต้นด้วย <? ตามด้วยคำสั่งที่เรียกฟังก์ชั่นหรือข้อความ และปิดท้ายด้วย ?> สำหรับตัวอย่างนี้เป็นสคริปต์ที่แสดงข้อความว่า "Hi, I'm a PHP script" โดยใช้คำสั่ง echo ซึ่งเป็นคำสั่งที่ใช้ในการแสดงผลของภาษาสคริปต์ PHP ซึ่งจะแสดงผลดังนี้ http://www.mwit.ac.th/~jeab/40201/pic/click_here.gif |
| http://www.mwit.ac.th/~jeab/40201/pic/goo_word_icon.gif เราสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจหนึ่งๆ โดยเปิดและปิดด้วยแท็ก(Tag) ของ PHP กี่ครั้งก็ได้ ดังตัวอย่างต่อไปนี้ |
| ตัวอย่างที่ 2 |
|

|  |  |
| --- | --- |
| 1234567891011121314151617 | <html> <head> <title>Example 1 </title> </head> <body><table border=1><tr><td>**<? echo"PHP script block 1"; ?>**</td><td>**<? echo"PHP script block 2 "; ?>**</td></tr></table>**<?    echo"PHP script block 3 <br> ";   echo date("ขณะนี้เวลา H:i น."); ?>** </body> </html> |

แสดงผลลัพธ์  |
| **ความสามารถของภาษา PHP** |
| * เป็นภาษาที่มีลักษณะเป็นแบบ Open source ผู้ใช้สามารถ Download และนำ Source code ของ PHP ไปใช้ได้โดยไม่เสียค่าใช้จ่าย
* เป็นสคริปต์แบบ Server Side Script ดังนั้นจึงทำงานบนเว็บเซิร์ฟเวอร์ ไม่ส่งผลกับการทำงานของเครื่อง Client โดย PHP จะอ่านโค้ด และทำงานที่เซิร์ฟเวอร์ จากนั้นจึงส่งผลลัพธ์ที่ได้จากการประมวลผลมาที่เครื่องของผู้ใช้ในรูปแบบของ HTML ซึ่งโค้ดของ PHP นี้ผู้ใช้จะไม่สามารถมองเห็นได้
* PHP สามารถทำงานได้ในระบบปฎิบัติการที่ต่างชนิดกัน เช่น Unix, Windows, Mac OS หรือ Risc OS อย่างมีประสิทธิภาพ เนื่องจาก PHP เป็นสคริปต์ที่ต้องทำงานบนเซิร์ฟเวอร์ ดังนั้นคอมพิวเตอร์สำหรับเรียกใช้คำสั่ง PHP จึงจำเป็นต้องติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์ไว้ด้วย เพื่อให้สามารถประมวลผล PHP ได้
* PHP สามารถทำงานได้ในเว็บเซิร์ฟเวอร์หลายชนิด เช่น Personal Web Server(PWS), Apache, OmniHttpd และ Internet Information Service(IIS) เป็นต้น
* ภาษา PHP สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)
* PHP มีความสามารถในการทำงานร่วมกับระบบจัดการฐานข้อมูลที่หลากหลาย ซึ่งระบบจัดการฐานข้อมูลที่สนับสนุนการทำงานของ PHP เช่น Oracle, MySQL, FilePro, Solid, FrontBase, mSQL และ MS SQL เป็นต้น
* PHP อนุญาตให้ผู้ใช้สร้างเว็บไซต์ซึ่งทำงานผ่านโปรโตคอลชนิดต่างๆ ได้ เช่น LDAP, IMAP, SNMP, POP3 และ HTTP เป็นต้น
* โค้ด PHP สามารถเขียน และอ่านในรูปแบบของ XML ได้
 |

**หลักการทำงานของ PHP**



**ภาพที่ 3** แสดงขั้นตอนการทำงาน PHP Script Request/Respons

1) จากไคลเอนต์จะเรียกไฟล์ php script ผ่านทางโปรแกรมบราวเซอร์ (Internet Explore)

2) บราวเซอร์จะส่งคำร้อง (Request) ไปยังเว็บเซิร์ฟเวอร์ผ่านทางเครือข่ายอินเทอร์เน็ต

3)   เมื่อเว็บเซิร์ฟเวอร์รับคำรองขอจากบราวเซอร์แล้วก็จะนำสคริปต์phpที่เก็บอยู่ในเซิร์ฟเวอร์มาประมวลผลด้วยโปรแกรมแปลภาษา PHP ที่เป็นอินเตอร์พรีเตอร์

4)  กรณีที่php script มีการเรียกใช้ข้อมูลก็จะติดต่อกับฐานข้อมูลต่างๆผ่านทาง ODBC Connection ถ้าเป็นฐานข้อมูลกลุ่ม Microsoft SQL Server, Microsoft Access, FoxProหรือใช้Function Connection ที่มีอยู่ใน PHP Library ในการเชื่อมต่อฐานข้อมูลเพื่อดึงข้อมูลออกมาหลังจากแปลสคริปต์ PHP เสร็จแล้วจะได้รับไฟล์ HTML ใหม่ที่มีแต่แท็กHTML ไปยัง Web Server

5)  Web Server ส่งไฟล์ HTML ที่ได้ผ่านการแปลแล้วกลับไปยังบราวเซอร์ที่ร้องขอผ่านทางเครือข่ายอินเตอร์เน็ต

6)  บราวเซอร์รับไฟล์ HTML ที่เว็บเซิร์ฟเวอร์ส่งมาให้แปล HTML แสดงผลออกมาทางจอภาพเป็นเว็บเพจโดยใช้ตัวแปลภาษา HTML ที่อยู่ในบราวเซอร์ซึ่งเป็นอินเตอร์พรีเตอร์เช่นเดียวกัน

รูปแบบการเขียน PHP

การเขียนโค้ด เราสามารถเขียนได้จากโปรแกรม Editor ทั่วไปเช่น Notepad หรือ Editplus แน่นอนที่สะดวกที่สุดคงจะไม่พ้น Notepad เพราะแถมมากับ window อยู่แล้ว แต่ถ้าต้องการความสามารถและ Options ที่เพิ่มขึ้นก็แนะนำว่าโปรแกรม Editplus ใช้ได้ดีทีเดียว

รูปแบบการเขียน PHP เขียนได้ 4 แบบดังตัวอย่าง ที่นิยมคือแบบที่ 1 และ 2 แบบที่ 3 ใช้งานคล้ายกับ Java script  ส่วนแบบที่ 4 ตัว tag <% จะเหมือนกับ ASP โดยเมื่อรันจะได้ผลลัพธ์เหมือนกัน และสามารถแทรกลงในส่วนของภาษา HTML ส่วนใดก็ได้

1.การเขียนโค้ดในรูปแบบภาษา SGML จะมีรูปแบบดังนี้

<?
คำสั่งในภาษา PHP ;
?>

2. การเขียนโค้ดเพื่อใช้ร่วมกับภาษา XHTML หรือ XML (แต่สามารถใช้ใน HTML แบบปกติได้) จะมีรูปแบบดังนี้

<?php
คำสั่งในภาษา PHP ;
?>

3. การเขียนโค้ดในรูปแบบ JavaScript จะมีรูปแบบดังนี้

<Script Language=”php”>
คำสั่งในภาษา PHP ;
</Script>

4. การเขียนโค้ดในรูปแบบ ASP จะมีรูปแบบดังนี้

<%
คำสั่งในภาษา PHP ;
%>

\* สำหรับรูปแบบที่ 4 จะใช้ได้กับ PHP 3.0.4 ขึ้นไป และจะต้องไปแก้ไฟล์ php.ini ในโฟลเดอร์ C:\WINDOWS เสียก่อนโดยให้ asp\_tags มีค่าเป็น On

การเขียนสคริปต์ PHP ในรูปแบบใดก็ตามจะต้องมีเครื่องหมาย semicolon ( ; ) ลงท้ายคำสั่งเสมอเหมือนกับการเขียนภาษา C กับภาษา Perl และคำสั่งหรือฟังก์ชั่นในภาษา PHP จะเขียนด้วยตัวพิมพ์เล็กหรือพิมพ์ใหญ่ก็ได้ ( case-insensitive ) การจบ statement หรือสิ้นสุด script เราจะปิดท้ายสคริปต์ด้วยแท็ก ( ?> ) และคำสั่งสุดท้ายในสคริปต์นั้นจะลงท้ายด้วย semicolon ( ; ) หรือไม่ก็ได้เพราะจะถูกปิดด้วยแท็ก ( ?> ) อยู่แล้ว

นอกจากรูปแบบแล้ว การวาง code ผสมกับ HTML ก็เป็นวิธีหนึ่ง

<html>
<head>
<title>Example</title>
</head>
<body>
<?php
echo “Hi, I’m a PHP script!”;
?>
</body>
</html>

Comment (การเขียนคำอธิบายโปรแกรม)

การเขียนโปรแกรมที่มีความยาวและซับซ้อนมากๆอาจจะทำให้สับสนในภายหลังได้ วิธีที่นิยมกันก็คือการเขียนคำอธิบายไว้ท้ายคำสั่งนั้นๆ หรือที่เรียกกันว่า comments ใน PHP จะสามารถเขียนในรูปแบบของภาษา C, C++ และ Unix shell-style comments ได้โดยจะไม่นำมาประมวลผล จะเห็นแค่ใน souce code เท่านั้น

รูปแบบ

<?php

echo “This is a test”;         // comment  แบบ C++

/\* แบบนี้เป็นการ comments
แบบหลายบรรทัด จะใช้ในกรณี
ที่คำอธิบายเยอะ\*/

echo “This is yet another test”;

echo “One Final Test”;      # comment แบบ Unix shell-style

?>

ข้อควรระวัง PHP ไม่รับ Comment แบบ nest

<?php

/\*

echo “This is a test”; /\* comment ตัวนี้จะมีปัญหา \*/

\*/

?>

คำสั่งแสดงผล

เราสามารถใช้คำสั่งเพื่อแสดงผลได้ 3 แบบคือ
1. echo
2. print
3. printf

1. คำสั่ง Echo จะสามารถแสดงได้หลายประเภท เช่น

<?php
echo ” ทดสอบการใช้คำสั่ง echo “;
?>

นี่เราลองมาดูความสามารถอีกอย่างของคำสั่ง echo กันคือความสามารถในการแยกนิพจน์ หรือค่าตัวแปรได้ โดยจะใช้เครื่องหมาย , คั่น

<?php
echo ” ทดสอบการใช้คำสั่ง echo<br> ” ;
echo ” <b>10+20 = ” , 15+15 , “</b>” ;
?>

สังเกตคำสั่ง echo “<b> 10+20 = ” , 15+15 , “</b>” ; ผมได้ใช้เครื่องหมาย , คั้นระหว่าง “<b> 10+20 =” และ “</b>” ไว้เพื่อให้โปรแกรมแยกส่วนที่เราต้องการให้มันแสดงออกทางหน้าแบบธรรมดากับส่วนที่เราต้องการให้โปรแกรมทำการคำนาณให้เรานั้นคือ 15+15 เมื่อคำนวณแล้วจะได้ค่า 30 โปรแกรมจะนะค่าที่ได้จากการคำนวณมาแสดงแทน ส่วนแท็ก <br> และ <b>…</b> นั้นเป็นแท็ก HTML ธรรมดาซึ่งผมใส่ไว้เพื่อทำให้การแสดงผลสวยงามขึ้น

<?php
echo “ทดสอบการใช้คำสั่ง echo ” ;
echo ” 10+20 = ” , 15+15 ;
?>

2. คำสั่ง print
<?php
print ” ทดสอบการใช้คำสั่ง print ” ;
?>

3. คำสั่ง Printf

ในการใช้คำสั่ง printf เราจะต้องทราบชนิดของข้อมูลที่เราต้องการแสดงออกมาว่าเป็นชนิดใด เราจะได้กำหนดค่าลงไปถูงต้องดังนี้

%d     ตัวเลข

%o       เลขฐานแปด

%c       ข้ออักษร ( 1 ตัว )

%s       ข้อความ

%f       ทศนิยม

<?php
printf ( ” 15+15 = %d <br> ” , 15+15) ;
printf ( ” 20/3 = %d <br> ” , 20/3 ) ;
printf ( ” 20/3 = %f <br> ” , 20/3 ) ;
?>

สังเกตคำสั่งที่ 2 และ 3 ให้ดีนะครับ เราได้ใช้ตัวคำนวณเหมือนกันแต่กำหนดชนิดของข้อมูลไม่เหมือนกัน โดยคำสั่งที่ 2 ผมได้กำหนดชนิดข้อมูลเป็น %d แต่ในคำสั่งที่ 3 ได้กำหนดชนิดเป็น %f ผลที่ได้ก็จะแตกต่างการกันครับ

String

แบ่งตามลักษณะตัวปิดแบ่งออกเป็น 3 แบบคือ

• single quoted

• double quoted

• heredoc syntax (ไม่อธิบาย)

Single Quoted

ตัวแปร ที่อยู่ภายใต้ single quoted ถือเป็นข้อความด้วย

echo ’this is a simple string’;

echo ’You can also have embedded newlines in strings,

like this way.’;

echo ’Arnold once said: “I\’ll be back”’;   // output: … “I’ll be back”

echo ’Are you sure you want to delete C:\\\*.\*?’;      // output: … delete C:\\*.\*?

echo ’Are you sure you want to delete C:\\*.\*?’;        // output: … delete C:\\*.\*?

echo ’I am trying to include at this point: \n a newline’; // output: … this point: \n a newline

Double Quoted

การใช้ double quoted สามารถใช้ร่วมกับ escaped characters ได้ดังตาราง

Escaped characters sequence meaning

ข้อควรระวังในการใช้ งาน

$beer = ’Heineken’;

echo “$beer’s taste is great”; // works, “’” is an invalid character for varnames

echo “He drunk some $beers”; // won’t work, ’s’ is a valid character for varnames

echo “He drunk some ${beer}s”; // works

Simple syntax

$fruits = array( ’strawberry’ => ’red’ , ’banana’ => ’yellow’ );

echo “A banana is $fruits[banana].”;

echo “This square is $square->width meters broad.”;

echo “This square is $square->width00 centimeters broad.”; // won’t work,

// for a solution, see the complex syntax.

Complex syntax

$great = ’fantastic’;

echo “This is { $great}”; // won’t work, outputs: This is { fantastic}

echo “This is {$great}”; // works, outputs: This is fantastic

echo “This square is {$square->width}00 centimeters broad.”;

echo “This works: {$arr[4][3]}”;

echo “This is wrong: {$arr[foo][3]}”; // for the same reason

// as $foo[bar] is wrong outside a string.

echo “You should do it this way: {$arr[’foo’][3]}”;

echo “You can even write {$obj->values[3]->name}”;

echo “This is the value of the var named $name: {${$name}}”;

ตัวอย่างการใช้งาน String

<?php

$str = “This is a string”;    /\* การกำหนดค่าให้กับ string. \*/

$str = $str . ” with some more text”;        /\* ต่อข้อความกับตัวแปร \*/

$str .= ” and a newline at the end.\n”; /\* ต่อข้อความกับตัวแปร อีกรูปแบบหนึ่ง และใช้ escaped newline. \*/

/\* This string will end up being ’<p>Number: 9</p>’ \*/

$num = 9;+

$str = “<p>Number: $num</p>”;

/\* This one will be ’<p>Number: $num</p>’ \*/

$num = 9;

$str = ’<p>Number: $num</p>’;

/\* Get the first character of a string \*/

$str = ’This is a test.’;

$first = $str{0};

/\* Get the last character of a string. \*/

$str = ’This is still a test.’;

$last = $str{strlen($str)-1};

?>

Variable Scope

PHP โดยส่วนใหญ่ตัวแปรจะเป็นแบบ Single scope ดังแสดงตามตัวอย่าง

$a = 1;

include “b.inc”;

ตัวอย่าง การใช้ตัวแปร global และ local

แบบที่1 ตัวแปร a มีค่าต่างกัน

$a = 1; /\* global scope \*/

Function Test () {

echo $a; /\* reference to local scope variable \*/

}

Test ();

แบบที่ 2 การใช้ตัวแปร a และ b

$a = 1;

$b = 2;

Function Sum () {

global $a, $b;

$b = $a + $b;

}

Sum ();

echo $b;

HTML FORM

<form action=”foo.php” method=”post”>

Name: <input type=”text” name=”username”><br>

<input type=”submit”>

</form>

pass by reference

function add\_some\_extra(&$string) {

$string .= ’and something extra.’;

}

$str = ’This is a string, ’;

add\_some\_extra($str);

echo $str; // outputs ’This is a string, and something extra.’

PostgreSQL ใน PHP

int pg\_connect (string host, string port, string options, string tty, string dbname)

ตัวอย่าง

$dbconn3 = pg\_Connect (“host=sheep port=5432 dbname=mary user=lamb password=baaaa”);

pg\_close

bool pg\_close (int connection)

pg\_cmdtuples

Returns number of affected tuples

int pg\_cmdtuples (int result\_id)

pg\_exec

Execute a query

int pg\_exec (int connection, string query)

pg\_result

Returns values from a result identifier

mixed pg\_result (int result\_id, int row\_number, mixed fieldname)

pg\_freeresult

Free result memory

int pg\_freeresult (int result\_id)

ตัวอย่างการใช้งาน

<?

function getAuthorID($dbconn) {

$sql = “select nextval(‘authors\_author\_id\_seq’)”;

$result = pg\_exec($dbconn,$sql);

return pg\_result($result,0,0);

}

function insertAuthor($dbconn,$aid,$title,$fname,$lname,$mail,$usr) {

$sql = “insert into authors (author\_id, author\_title, author\_fname, author\_lname, author\_email, a\_user\_name) values (‘$aid’, ‘$title’, ‘$fname’, ‘$lname’, ‘$mail’, ‘$usr’) “;

$result = pg\_exec($dbconn,$sql);

$err =  pg\_cmdtuples($result);

return $err;

}

$sql = “SELECT paper\_id,trim(paper\_title) as p\_title, trim(abstract) as p\_abstract,

FROM papers

WHERE paper\_id = ‘$p\_paper\_id’ “;

$row = 0;

$rs = pg\_exec($km\_connect,$sql);

$p\_title = pg\_result($rs,$row,p\_title);

$p\_abstract = pg\_result($rs,$row,p\_abstract);       ?>