

ใบความรู้ที่ 1

เรื่องตัวแปร และ ตัวดำเนินการ

การเขียนโปรแกรม PHP ร่วมกับ HTML

การเขียนโปรแกรมในภาษา PHP นั้น จะต้องเขียนลงไปอยู่ใน file HTML (หรือว่า file PHP) ที่ต้องการให้แสดงผล และเนื่องจาก PHP เป็น server-side script จึงทำให้ browser ไม่สามารถเห็น PHP code ได้ มีเพียง web server ที่เห็นและทำการประมวลผล PHP code ก่อนข้อมูลจะไปแสดงผลที่ web server สามารถแยกแยะระหว่าง PHP code และ HTML code ได้ จึงต้องมีการกำหนดขอบเขตว่า ส่วนไหนคือส่วนของ HTML และส่วนไหนคือส่วนของ PHP ซึ่ง PHP มี tag พิเศษที่ใช้กำหนดขอบเขตได้ 4 แบบ คือ

- นำหน้าด้วย `<?` และปิดท้ายด้วย `?>` เช่น

```
<? Echo ("this is SGML-style PHP escaping tag")?>
```

- นำหน้าด้วย `<?php` และปิดท้ายด้วย `?>` เช่น

```
<?php echo ("this is SGML-style PHP escaping tag")?>
```

- นำหน้าด้วย `<%` และปิดท้ายด้วย `%>` เช่น

```
<% echo ("this is SGML-style PHP escaping tag")%>
```

- นำหน้าด้วย `<?script langauge="php">` และปิดท้ายด้วย `</script>` เช่น

```
<script>
    echo ("This is Standard HTML scripting tag")
</script>
```

ซึ่งใน HTML file เดียวกันนั้น อาจจะใช้ tag เหล่านี้ปนกันได้ แต่แนะนำว่า ให้ใช้เพียงแบบเดียว เพราะจะทำให้ง่ายต่อการเขียน และการค้นหาข้อผิดพลาด นอกจากนั้นยังสามารถแทรก phpcode ได้ทุกตำแหน่งใน HTML file อีกด้วย

การเขียน PHP script ใน HTML file

```
<H1>Example1</H1>

<P align=center>

<?
Print "Hello world";
?>
```

ตัวอย่างที่ 1 การเขียน PHP script แสดงผลออกทาง web browser

PHP script ที่ Web server เห็น	PHP script ที่ Web server ได้รับ
<H1>Example1</H1>	<H1>Example1</H1>
<P align=center>	<P align=center>
<?	Hello world
Print "Hello world";	</P>
?>	
</P>	

การเขียน comment

เพื่อการเขียนชุดคำสั่ง PHP อย่างมีประสิทธิภาพ การเขียนคำแนะนำ (comment) ไว้ในชุดคำสั่งจะทำให้ผู้อื่นรวมทั้งผู้เขียนสามารถเข้าใจชุดคำสั่งได้ง่ายขึ้น วิธีการเขียน comment ลงไปใน php code ใช้วิธีการเหมือนกับภาษา C และ Perl คือ ใช้สัญลักษณ์ `/* */` หรือ `//` หรือ `#`

- `/*` (เริ่มต้น comment)

คุณสามารถเขียน comment ได้หลายบรรทัด

`*/` (สิ้นสุด comment)

```
<?
    /* This is my first script
       It prints "This is PHP" to Web page
    */
```

- `//` หรือ `#` เขียน comment ได้บรรทัดเดียว และไม่มีการสิ้นสุด comment

```
<? // This is my first script
    # It prints "This is PHP" to Web page
    echo "This is PHP<br>\n";
?>
```

การแยกคำสั่ง

ในกรณีที่ต้องการเขียนคำสั่งของ php จะต้องมีการแยกคำสั่งแต่ละคำสั่งออกจากกัน ซึ่งจะใช้เครื่องหมาย ; หรือว่า semi colon เป็นตัวแยก ดังตัวอย่าง

```
<?
Echo ("This is the first command");
Echo "This is the second command";
```

ซึ่ง จะให้ผลการทำงานเช่นเดียวกับชุดคำสั่งในรูปแบบนี้

```
<? Echo ("This is the first command"); Echo "This is the second command"; ?>
```

ข้อแนะนำคือ ควรจะเขียน 1 คำสั่งต่อ 1 บรรทัด เพราะจะทำให้อ่านง่าย และตรวจสอบ แก้ไขได้ง่าย และขนาด file ที่ใหญ่ หรือยาว ขึ้น ไม่มีผลต่อความเร็วในการทำงานเท่าไรนัก

ตัวแปร (Variable) และชนิดของตัวแปร (Variable type)

ภาษา PHP ก็เหมือนกับภาษาโปรแกรมโดยทั่วไปที่จะต้องมิตัวแปรเพื่อใช้เก็บค่า (Value) การกำหนดชื่อของตัวแปรใน PHP จะใช้สัญลักษณ์ \$ นำหน้าชื่อตัวแปร (ชื่อตัวแปรเป็นแบบ case sensitive กล่าวคือ A กับ a ถือว่าเป็นคนละตัวกัน) เช่น กำหนดค่า 100 ให้กับตัวแปรชื่อ \$price ต้องเขียนดังนี้

```
$price = 100;
```

ส่วนชนิดของตัวแปรในภาษา PHP จะมีตัวแปรทั้งหมด 7 ชนิดคือ

- integer ใช้สำหรับเก็บข้อมูลชนิดเลขจำนวนเต็ม
- double ใช้สำหรับเก็บข้อมูลชนิดเลขทศนิยม
- string ใช้สำหรับเก็บข้อมูลตัวอักษร หรือว่า กลุ่มของตัวอักษร
- array ใช้สำหรับเก็บกลุ่มข้อมูล
- object เป็นชนิดข้อมูลสำหรับการเขียนโปรแกรมแบบ Object Oriented
- pdfdoc ใช้เก็บเอกสารในรูปแบบ PDF (ต้องเลือกให้ PHP สนับสนุน PDF ในขณะติดตั้ง ถึงจะใช้ได้)
- pdfinfo ใช้เก็บข้อมูลเกี่ยวกับเอกสาร PDF (ต้องเลือกให้ PHP สนับสนุน PDF ในขณะติดตั้ง ถึงจะใช้ได้)

ในภาษา PHP นั้น ไม่จำเป็นต้องประกาศ ชนิดของตัวแปรก่อนที่จะใช้งาน โดย PHP จะตรวจสอบเองว่า ข้อมูลที่เก็บในตัวแปรนั้น เป็นข้อมูลชนิดใด และก็จะกำหนดชนิดของตัวแปรให้เอง ตัวอย่างเช่น

```
$price = 100;
```

```
$firstName = "Thanya";
```

```
$nickName = "Oh";
```

```
$name = "$firstName";
```

บรรทัดที่ 1 : PHP เข้าใจว่า 100 เป็น integer → \$price เป็นตัวแปรชนิด integer

บรรทัดที่ 2 : PHP เข้าใจว่า Thanya เป็น string เนื่องจากมีสัญลักษณ์ “ (double quotation) → \$firstName เป็น string

บรรทัดที่ 3 : PHP เข้าใจว่า Oh เป็น string เนื่องจากมีสัญลักษณ์ ‘ (single quotation) → \$nickName เป็น string

บรรทัดที่ 4 : ตัวแปร \$name ถูกกำหนดค่าให้เท่ากับ \$firstName → ชนิดของตัวแปร \$name เหมือนกับ \$firstName คือเป็น string นั่นเอง

ข้อสังเกต

การเขียน `$name = "$firstname"` จะได้ค่า `$name = "Thanya"` เนื่องจากตัวแปรที่อยู่ภายใต้เครื่องหมาย “ จะถูกประมวลผลก่อนนำไปใช้

แต่การเขียน `$name = "$firstname"` จะได้ค่า `$name = "$firstname"` เนื่องจากตัวแปรที่อยู่ภายใต้เครื่องหมาย ‘ จะไม่ถูกประมวลผลก่อนนำไปใช้

Operations and Comparison

การเขียนโปรแกรมโดยทั่วไปย่อมต้องมีการคำนวณค่าของตัวแปรในแบบต่างๆ ด้วย ไม่ว่าจะเป็นการคำนวณเชิงคณิตศาสตร์หรือเชิงเปรียบเทียบก็ตาม PHP มีกลุ่มของ Operator ที่ช่วยในการคำนวณต่างๆ แบ่งได้เป็น

1. Assignment Operator

เป็น Operator ที่ใช้ในการกำหนดค่า ซึ่งเราได้ทดลองใช้ไปแล้ว คือ เครื่องหมาย = (เท่ากับ) นั้นเอง เช่น `$val1=5;`

2. Arithmetic Operators

เป็น Operator ที่ใช้เกี่ยวกับการคำนวณทางคณิตศาสตร์ สามารถใช้กับข้อมูลที่มีชนิดเป็นตัวเลขเท่านั้นซึ่งสามารถสรุปได้เป็นตารางดังนี้ โดยสมมติให้ `$a=5` และ `$b=12`

Expression	ชื่อ	การทำงาน	ผลลัพธ์
<code>\$a + \$b</code>	Addition	รวมค่าข้อมูลใน <code>\$a</code> และ <code>\$b</code>	17
<code>\$a - \$b</code>	Subtraction	นำค่าข้อมูลใน <code>\$b</code> ไปลบออกจาก <code>\$a</code>	-7
<code>\$a * \$b</code>	Multiplication	คูณค่าข้อมูลใน <code>\$a</code> และ <code>\$b</code>	60
<code>\$a / \$b</code>	Division	หารข้อมูลใน <code>\$a</code> ด้วย <code>\$b</code>	2.4

\$a % \$b	Modulus	หาค่ามอดุลัส (เศษการหาร) ของ \$a และ \$b	2
-----------	---------	--	---

3. String Operators

Operation ที่สามารถกระทำกับ string ได้คือการต่อ String (Concatenation) String ซึ่ง Operation ใน PHP ก็มีเพียงแค่ 1 Operator เท่านั้น คือ Concatenation Operator โดยใช้เครื่องหมาย . (dot) ตัวอย่างเช่น

```
$firstName="Thanyaluk";
```

```
$fullName=$firstName."Jirapech-umpai";
```

```
จะได้ $fullName="ThanyalukJirapech-umpai";
```

หากต้องการเว้นช่องว่างก็ให้พิมพ์

```
$fullName=$firstName." Jirapech-umpai";
```

```
จะได้ $fullName="Thanyaluk Jirapech-umpai";
```

4. Execution Operations

ใช้สำหรับ execute shell command โดย PHP จะเอาสิ่งที่อยู่ในเครื่องหมาย ` (backticks) มาทำการ execute ตัวอย่าง เช่น

```
$output = ` dir `;
```

```
echo "<pre>$output</pre>"; (ผลลัพธ์ที่ได้จะเหมือนคำสั่ง dir ใน command prompt)
```

5. Comparison Operations

ใช้ในการเปรียบเทียบค่าต่างๆ ซึ่งจะให้ผลลัพธ์เพียงเป็นจริง (TRUE) หรือว่าเป็นเท็จ (FALSE) เท่านั้น สามารถสรุปเป็นตารางได้ดังนี้

Expression	ชื่อ	การทำงาน
$a == b$	Equal	เป็นจริงถ้า a เท่ากับ b
$a != b$	Not equal	เป็นจริงถ้า a ไม่เท่ากับ b
$a < b$	Less than	เป็นจริงถ้า a น้อยกว่า b
$a > b$	Greater than	เป็นจริงถ้า a มากกว่า b
$a <= b$	Less than or equal to	เป็นจริงถ้า a น้อยกว่า หรือ เท่ากับ b
$a >= b$	Greater than or equal to	เป็นจริงถ้า a มากกว่า หรือ เท่ากับ b

6. Logical Operations

ใช้เปรียบเทียบเชิงตรรกะ หรือเปรียบเทียบข้อเท็จจริง นั้นเอง สามารถสรุปได้เป็นตารางได้ดังนี้

OPERATOR	ชื่อ	การทำงาน	EXPRESSION	ผลลัพธ์
AND	AND	เป็นจริงถ้า Expression ทั้งซ้ายและขวา เป็นจริง	$(5 > 2)$ and $(3 > 2)$ $(2 > 5)$ and $(2 > 3)$	True False
OR	OR	เป็นจริงถ้า Expression ด้านซ้ายหรือ ด้านขวาเป็นจริง	$(5 > 2)$ or $(2 > 7)$ ("h" > "a") or $(1 < 3)$	True True
XOR	Exclusiv	เป็นจริงถ้า Expression ด้านซ้ายหรือ	("ar" > "ac") xor	True

	e Or	ด้านขวาใดด้านหนึ่งเท่านั้นเป็นจริง	(9>2) (3 == 3) xor ("f" == "f")	False
!	Not	เป็นจริงถ้า Expression เป็นเท็จ เป็นจริงถ้า Expression เป็นจริง	!(5<10)	False