PHP (HyperText Preprocessor)

<u>บทนำ</u>

PHP (Hypertext Preprocessor) เป็น server side scripting language ซึ่งมีโครงสร้างของภาษา คล้าย C , Java และ Perl ถูกคิดค้นขึ้นโดยนาย Rasmus Lerdorf ซึ่งจุดประสงค์ของเขาเพื่อที่จะสร้าง code ที่ซับซ้อนใส่เข้าไปใน HTML ให้ได้ การเขียน php script นั้นไม่จำเป็นต้องประกาศตัวแปรก่อนการใช้งาน นอกจากนั้น PHP ยังรองรับการเขียนโปรแกรมแบบ Object-Oriented ได้อีกด้วย ในปัจจุบัน PHP จึงกลายเป็น scripting language ที่ได้รับความนิยมและมีความสามารถสูงภาษาหนึ่ง

PHP เป็น scripting language ที่ถูกสร้างขึ้นมาสำหรับงานด้าน Web Application ซึ่งมีความสามารถด้าน Database เป็นความสามารถหลักและได้รับการยอมรับว่า เป็น Web Application language ที่มีความเร็วสูงที่สุด มีประสิทธิภาพมากที่สุด และพัฒนาได้ง่ายที่สุดภาษาหนึ่ง เนื่องจาก PHP มี build-in function ให้นักพัฒนาโปรแกรมเลือกใช้เป็นจำนวนมากในปัจจุบัน PHP ได้พัฒนาถึง Version 4

ประวัติของ PHP

- Conceived in fall of 1994 to track people looking at his online resume.
- PHP Version 1 in spring 1995
- PHP Version 2 1995-1997
- PHP Version 3 1997-now
- PHP Version 4 Q4 1999

ผู้พัฒนา

- Zeev Suraski and Andi Gutmans in Israel
- Shane Caraveo in Florida
- Stig Bakken in Norway
- Andrey Zmievski in Nebraska
- Sascha Schumann in Dortmund, Germany
- Thies C. Arntzen in Hamburg, Germany
- Jim Winstead in Los Angeles
- Rasmus Lerdorf in North Carolina

ไวยากรณ์ (SYNTAX) ของ PHP

1. การเขียนโปรแกรม PHP ร่วมกับ HTML

การเขียนโปรแกรมในภาษา PHP นั้น จะต้องเขียนลงไปใน 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  language=“php”>
        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

ตัวอย่าง การเขียน PHP script ใน HTML file

```
<H1>Example1</H1>
<P align=center>
<?
Print “Hello world”;
?>
```

ตัวอย่าง การแสดงผล PHP script ใน HTML file ออกทาง web browser

HTML ที่ Web server	HTML ที่ Web browser ได้รับ
<H1>Example1</H1> <P align=center> <?	<H1>Example1</H1> <P align=center> Hello world

Print “Hello world”; ?> </P>	</P>

2. <u>การเขียน comment</u>

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

- /* (เริ่มต้น comment)

  สามารถเขียน comment ได้หลายบรรทัด

  */ (สิ้นสุด comment)

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

- // หรือ # เขียน comment ได้เพียงบรรทัดเดียว โดยจะสิ้นสุด comment เมื่อสิ้นสุดบรรทัด

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

3. <u>การแยกคำสั่ง</u>

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

4. <u>ตัวแปร (Variable) และชนิดของตัวแปร (Variable type)</u>

ภาษา PHP ก็เหมือนกับภาษาโปรแกรมโดยทั่วไปที่จะต้องมีตัวแปรเพื่อใช้เก็บค่า (Value) การกำหนดชื่อของตัวแปรใน PHP จะใช้สัญลักษณ์ $ นำหน้าชื่อตัวแปร (ชื่อตัวแปรเป็นแบบ case sensitive) 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 นั้นไม่จำเป็นต้องประกาศ ชนิดของตัวแปรก่อนที่จะใช้งานโดย PHP จะตรวจสอบเองว่า ข้อมูลที่เก็บในตัวแปรนั้น เป็นข้อมูลชนิดใด และก็จะกำหนดชนิดของตัวแปรให้เอง ตัวอย่างเช่น

```
$price = 100;
$firstName = “Thanya”;
$nickName = “Oh”;
$name = “firstName”;
```

บรรทัดที่ 1 : 100 เป็น integer → $price เป็นตัวแปรชนิด integer

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

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

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

<u>ข้อสังเกต</u>

การเขียน $name = “$firstname” จะได้ค่า $name = “Thanya” เนื่องจากตัวแปรที่อยู่ภายใต้เครื่องหมาย “ จะถูกประมวลผลก่อนนำไปใช้

แต่การเขียน $name = ‘$firstname’ จะได้ค่า $name = ‘$firstname’ เนื่องจากตัวแปรที่อยู่ภายใต้เครื่องหมาย ‘ จะไม่ถูกประมวลผลก่อนนำไปใช้

5. ตัวแปร Array

เป็นตัวแปรที่ใช้เก็บกลุ่มค่าของข้อมูล (“series or collection of things”) เช่น การเก็บชุดค่าของสีต่างๆ ไว้ในตัวแปรเดียวกัน คือการเก็บค่า scalar หลายๆ ค่าไว้ด้วยกัน (“blue”,”yellow”,”red”,etc.) ตัวแปรที่สามารถเก็บชุดข้อมูล scalar ไว้รวมกันได้ จะถูกเรียกว่า array

ตัวแปร array ใน PHP เป็นได้ทั้ง indexed array และ associative array

5.1 indexed array

การกำหนดค่าให้ array สามารถใช้ array() function ดังนี้

```
$colors=array(“blue”,”yellow”,”brown”);          //สำหรับสร้างตัวแปร array ที่ทราบค่าอยู่แล้ว
```

(หรือ $colors = array(); // สำหรับสร้างตัวแปร array ที่ยังไม่ทราบค่า)

การกำหนดค่าให้ array แบบธรรมดาสามารถทำได้ดังนี้

```
$colors[] =”green” ;  //ใช้กำหนดค่าให้กับตัวแปร array ที่ชื่อ $colors
```

ในขณะนี้ ค่าตัวแปร $colors มีทั้งหมด 4 ค่า คือ “blue”, ”yellow”, ”brown”, ”green” ส่วนการอ้างอิงค่าแต่ละค่าที่อยู่ใน array จะต้องใช้ key เพื่ออ้างอิง ในที่นี้ไม่มีการสร้าง key ขึ้นมาเลย PHP จึงใช้วิธีกำหนดตัวเลขขึ้นมาเป็น key โดยให้

key สำหรับ ค่าแรก (“blue”) คือ 0 → วิธีการเขียนตัวแปรเพื่ออ้างอิงค่าคือ $colors[0]

key สำหรับ ค่าที่ 2 (“yellow”) คือ 1 → วิธีการเขียนตัวแปรเพื่ออ้างอิงค่าคือ $colors[0]

เป็นลำดับเช่นนี้เรื่อยๆ ไป

แต่ข้อเสียของวิธีอ้างอิงค่าโดยใช้ตัวเลขเป็น key คือ key เหล่านี้ไม่มีความหมาย เช่น เมื่ออ้างอิง $colors[2] เราไม่สามารถทราบได้เลยว่าเลข 2 หมายถึงอะไร มีความเกี่ยวข้องกับสีแดงอย่างไร จึงทำให้มีการกำหนด key แบบใหม่เกิดขึ้นที่เรียกว่า associative array นั่นเอง

5.2 associative array

การสร้าง key ที่มีความหมาย เป็นการกำหนด key ให้เป็นมากกว่าตัวเลข ตัวอย่างเช่น array $colors ข้างต้น ซึ่งสีที่อยู่ใน $colors ทั้งหมดหมายถึงสีของแต่ละส่วนของห้องทำงาน จึงควรจะกำหนด key ให้แต่ละสีโดยใช้องค์ประกอบของห้องทำงานดังนี้

ม่าน(curtain) = blue

ประตู(door) = yellow

โต๊ะ(table) = brown

ชั้นวางของ(shelf) = green

วิธีการสร้าง associative array ทำได้ดังนี้

```
$colors = array(     “curtain”  =>  “blue”,
                     “door”     =>  “yellow”,
                     “table”    =>  “brown”,
                     “self”     =>  “green”);
```

วิธีอ้างอิงค่าใน associative array ทำได้ดังนี้

```
print $colors[door];   // หากต้องการพิมพ์สีของประตู ซึ่งก็คือสีเหลืองนั่นเอง
```

6. ตัวแปร และการสร้างฟอร์ม (Forms)

การเขียน server-side script นั้น ส่วนใหญ่แล้วเกิดจากความต้องการที่จะประมวลผลข้อมูลที่ได้จาก web page ไม่ว่าจะเป็นข้อมูลที่มาจากแบบฟอร์ม หรือ จาก link ใน web browser ก็ตาม PHP สามารถดึงข้อมูลจาก web มาใช้ได้โดยง่าย เนื่องจาก PHP จะทำการโอนย้ายชื่อ ตัวแปรใน php script โดยอัตโนมัติ สามารถดูได้จากตัวอย่างดังต่อไปนี้

```
<form action="process.php" method="get">
Please enter your e-mail address:
<input type=text size=20 name="email">
<br /><input  type=submit  value="submit">
</form>
```

เมื่อมีการกรอกข้อมูลลงในฟอร์มแล้วกดปุ่ม Submit

- ตัวแปรชื่อ email พร้อมกับค่าที่ผู้ใช้ป้อนเข้ามาจะถูกส่งไปยังไฟล์ process.php
- ในไฟล์ process.php จะมีการสร้างตัวแปรชื่อ email ที่มีค่าเหมือนกับค่าที่ผู้ใช้กรอกผ่าน web browser โดยอัตโนมัติ ดังนั้นเราจึงสามารถใช้ค่าจาก web form ได้ทันทีโดยใช้ชื่อตัวแปรเดียวกัน

ตัวอย่าง การเขียน PHP ในไฟล์ process.php

```
<?
        Echo "Your address is $email ,Thank you very much";
?>
```

7. การดำเนินการ (Operations) และ การเปรียบเทียบ (Comparison)

การเขียนโปรแกรมโดยทั่วไปย่อมต้องมีการคำนวณค่าของตัวแปรในแบบต่างๆ ด้วย ไม่ว่าจะเป็นการคำนวณเชิงคณิตศาสตร์หรือเชิงเปรียบเทียบก็ตาม PHP มีกลุ่มของตัวดำเนินการ (Operator) ที่ช่วยในการคำนวณต่างๆ แบ่งได้เป็น

7.1 Assignment Operator

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

7.2 Arithmetic Operators

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

Expression	ชื่อ	การทำงาน	ผลลัพธ์
$a + $b	Addition	รวมค่าข้อมูลใน $a และ $b	17
$a - $b	Subtraction	นำค่าข้อมูลใน $b ไปลบออกจาก $a	-7
$a * $b	Multiplication	คูณค่าข้อมูลใน $a และ $b	60
$a / $b	Division	หารข้อมูลใน $a ด้วย $b	2.4
$a % $b	Modulus	หาค่ามอดูลัส (เศษการหาร) ของ $a และ $b	2

7.3 String Operators

การดำเนินการที่สามารถกระทำกับ string ได้คือการต่อ String (String Concatenation) ซึ่งใน PHP มีเพียง 1 ตัวดำเนินการ 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”;

7.4 Execution Operators

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

```
$output =` ls – al `;
echo “<pre>$output</pre>”;
```

7.5 Comparison Operators

ใช้ในการเปรียบเทียบค่าต่างๆ ซึ่งจะให้ผลลัพธ์เพียงเป็นจริง (TRUE) หรือว่าเป็นเท็จ (FALSE) เท่านั้น สามารถสรุปตัวดำเนินการเปรียบเทียบได้ดังนี้

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

7.6 Logical Operators

ใช้เปรียบเทียบเชิงตรรกะ หรือเปรียบเทียบข้อเท็จจริง นั่นเอง สามารถสรุปได้เป็นตารางได้ดังนี้

OPERATOR	ชื่อ	การทำงาน	EXPRESSION	ผลลัพธ์
AND	AND	เป็นจริงถ้า Expression ทั้งซ้ายและขวาเป็นจริง	(5>2) and (3>2) (2>5) and (2>3)	True False
OR	OR	เป็นจริงถ้า Expression ด้านซ้ายหรือด้านขวาเป็นจริง	(5>2) or (2>7) (“h”>”a”) or (1<3)	True True
XOR	Exclusive Or	เป็นจริงถ้า Expression ด้านซ้ายหรือด้านขวาด้านใดด้านหนึ่งท่านั้นเป็นจริง	(“ar”>”ac”) xor (9>2) (3 = = 3) xor (“f”= =”f”)	True False

!	Not	เป็นจริงถ้า Expression เป็นเท็จ เป็นจริงถ้า Expression เป็นจริง	!(5<10)	False

8. <u>Control Statement</u>

Control Statement เรียกได้อีกอย่างว่า “program flow” เป็นคำสั่งที่ทำหน้าที่ควบคุมการทำงานของแต่ละ statement ในโปรแกรม โดยปกติแล้ว statement จะถูกประมวลผลตามลำดับจากบรรทัดแรกจนถึงบรรทัดสุดท้าย Control statement ถูกสร้างขึ้นมาเพื่อใช้ควบคุมและเปลี่ยนแปลงลำดับของการประมวลผล statement เหล่านี้ เช่นอาจให้มีการประมวลผลบาง statement ในบางกรณีเท่านั้น Control Statement ทั้งหมดใน PHP มีดังนี้

8.1 IF

if เป็นโครงสร้างภาษาที่สำคัญอย่างหนึ่ง ใช้ในการตรวจสอบเงื่อนไขการทำงาน และกำหนดทิศทางในการทำงานที่เหมาะสมกับเงื่อนไขนั้น ซึ่งโครงสร้างของ if ก็เป็นเช่นเดียวกับภาษา C คือ

```
if (condition expression)
        Statement
```

โดย if จะตรวจสอบว่า เงื่อนไขใน expression เป็น จริง (TRUE) หรือไม่ ถ้าเป็นจริง ก็จะเข้าไปทำคำสั่งใน statement ถ้า ไม่เป็นจริง (FALSE) ก็จะไม่ทำคำสั่งใน statement ซึ่ง statement นี้ อาจจะเป็นคำสั่งเพียงคำสั่งเดียว หรือว่า เป็นชุดคำสั่งก็ได้ ซึ่งถ้าเป็นชุดคำสั่ง ก็ต้องมี วงเล็บปีกกา คร่อม ชุดคำสั่งนั้นด้วย จากตัวอย่าง

```
< ?
        …
        if ($a > $b)
                echo “ a is bigger than b”;
        if ($b > $c) {
                echo “b is bigger than c”;
                $c = $b;
        }
        …
? >
```

8.2 ELSE

ในกรณีที่ต้องการให้มีทางเลือก มากกว่าหนึ่งทาง โดยท่าเงื่อนไขเป็นจริง ก็ให้ทำทางเลือกหนึ่ง ถ้าไม่เป็นจริง ก็ให้ทำอีกทางเลือกหนึ่ง ก็สามารถทำได้โดยใช้ else เข้ามาช่วยใน if ดังนี้

```
        if (expression)
                true-statement;
        else
                false-statement;
```

<u>ตัวอย่าง</u>

```
< ?
```

```
        …
        if ($a > $b) {
                echo " a is bigger than b";
        } else {
                echo "a is not bigger than b";
        }
        …
? >
```

จากตัวอย่าง ถ้าเงื่อนไขเป็นจริง ก็จะทำคำสั่งชุดแรก ถ้าเงื่อนไขไม่เป็นจริง ก็จะไปทำใน else หรือว่า คำสั่งชุดที่สองแทน

8.3 ELSEIF

elseif เป็นการนำเอา else มารวมกับ if โดยเมื่อ เงื่อนไขแรกไม่เป็นจริง และต้องมาทำคำสั่งใน else ก็จะทำการตรวจสอบเงื่อนไขใน if ที่อยู่กับ else ทันที ซึ่งจะมีรูปแบบดังนี้

```
        If (expression–1)
                statement-1;
        elseif (expression-2)
                statement-2;
        else
                statement-3;
```

ตัวอย่าง

```
< ?
        …
        if ($a > $b) {
                echo "a is bigger than b";
        } elseif ($a == $b) {
                echo "a is equal to b";
        } else {
                echo "a is smaller than b";
        }
        …
? >
```

จากตัวอย่าง จะเห็นว่ามีการตรวจสอบเงื่อนไขว่า $a > $b หรือไม่ ถ้าเป็นจริง ก็จะทำคำสั่งในชุดแรก แล้วก็ออกจากเงื่อนไข if เลย แต่ถ้าไม่จริง ก็จะไปตรวจสอบเงื่อนไขที่สองคือ $a == $b หรือไม่ ถ้าเป็นจริง ก็จะไปทำคำสั่งในชุดที่สอง แต่ถ้าเป็นเท็จ ก็จะไปทำคำสั่งในชุดที่สามเลย

การใช้ elseif นั้น จะใช้ซ้อนกันกี่ชั้นก็ได้ ซึ่งการตรวจสอบเงื่อนไข ก็จะทำไล่มาเรื่อย ๆ

8.4 WHILE

while เป็นคำสั่งที่ทำให้เกิดการทำงานวนรอบ โดย while จะตรวจสอบเงื่อนไขการทำงานว่าเป็นจริงหรือไม่ ถ้าเป็นจริง ก็จะวนทำชุดคำสั่งที่กำหนดไว้ไปเรื่อย ๆ จนกว่าเงื่อนไขการทำงานจะเป็นเท็จ โครงสร้างของ while เป็นดังนี้

```
while (expression)
    statement
```

ตัวอย่าง

```
< ?
    …
    $i = 1;
    while ($i <= 10) {
        print $i;
        $i++;
    }
    …
? >
```

จากโปรแกรมตัวอย่างนี้ เริ่มต้นจากการกำหนดค่าตัวแปร $I ให้เป็น 1 ซึ่งเมื่อเข้ามายัง while ก็จะตรวจสอบว่า $I น้อยกว่าหรือเท่ากับ 10 หรือไม่ ซึ่งก็เป็นจริง ก็จะเข้าไปทำงานใน while ซึ่งจะแสดงค่า $I ออกมา แล้วทำการเพิ่มค่าขึ้นมาอีก 1 แล้วก็จะกลับไปตรวจสอยเงื่อนไขใน while ซึ่งก็จะทำงานเช่นนี้ไปเรื่อย ๆ จนกว่าการตรวจสอบเงื่อนไขใน while จะให้ค่าเป็นเท็จ ออกมา

8.5 DO…WHILE

do ..while จะคล้าย ๆ กับ while แต่จะต่างกันตรงที่ว่า while นั้น จะตรวจสอบเงื่อนไข ก่อนเข้าไปทำงาน แต่ว่า do..while จะทำงานก่อน แล้วค่อยตรวจสอบเงื่อนไข มีรูปแบบดังนี้

```
do
    statement
while (expression)
```

8.6 FOR

โดยทั่วไปแล้ว for จะถูกใช้งานต่างกับ while ตรงที่ว่า ไม่ได้เป็นการตรวจสอบเงื่อนไข แต่ใช้จำนวนครั้งในการทำงานเป็นการกำหนดการทำงานแทน โดย for จะมีโครงสร้างดังนี้

```
For (expr1; expr2; expr3)
    statement;
```

โดย expr1 จะถูกเรียกมาทำงาน เมื่อ for เริ่มทำงาน และจะทำงานแค่ครั้งเดียว และทุกครั้งก่อนที่เกิด loop ก็จะทำการตรวจสอบเงื่อนไขใน expr2 ว่าเป็นจริงหรือไม่ ถ้าเป็นจริง ก็จะเข้าไปทำคำสั่งใน statement แต่ถ้าไม่เป็นจริง

ก็จะออกจากการทำงานใน for ไป และหลังจากทำงานใน statement เสร็จแล้ว ก็จะมาทำงานใน expr3 ก่อน ที่จะไปตรวจสอบเงื่อนไขใน expr2 อีกครั้ง

8.7 BREAK

break ใช้หยุดการทำงานใน loop while, do..while และ for โดยที่ไม่ต้องทำการตรวจสอบเงื่อนไขของ loop เหล่านั้น ดูตัวอย่าง

```
< ?
        …
        $I = 0;
        while ( $i < 10) {
                if ($arr[$i] == “stop”) {
                        break;
                }
                $i++;
        }
        …
? >
```

จากตัวอย่าง ไม่ว่าค่า $i จะเป็นค่าเท่าไหร่ ถ้าข้อมูลใน array $arr[] เป็น stop เมื่อไหร่ ก็จะหยุดทำงานใน loop นี้ได้ทันที

8.8 CONTINUE

continue จะคล้าย ๆ กับ break นั่นคือ จะหยุดการทำงานใน loop ปัจจุบันเอาไว้ก่อน แต่จะต่างจาก break ตรงที่ว่า continue จะหลับไปเริ่มต้นทำงานใหม่ ที่ต้นของ loop แทนที่จะออกจาก loop ไปเลย

8.9 SWITCH

switch นั้น จะทำการตรวจสอบเงื่อนไข ที่ไม่เฉพาะ ถูกและผิด ได้ ซึ่งจะทำงานคล้าย ๆ กับ ชุดของ if / elseif นั่นเอง ต่างกันตรงที่ว่า switch จะเป็นการกำหนดตำแหน่งที่จะทำงานมากกว่า จะกำหนดชุดคำสั่งที่จะทำงาน มีรูปแบบดังนี้

```
        switch (expression) {
                case expr1 :
                        statement-1;
                case expr2 :
                        statement-2;
                …
        }
```

ตัวอย่าง

```
< ?
        …
        switch($I) {
                case 0 :
                        echo “$i equals 0”;
                        break;
                case 2 :
                        echo “$i equals 2”;
                        break;
        }
        …
? >
```

จากตัวอย่าง ถ้าค่า $i มีค่าเป็น 0 ก็จะเข้ามาทำที่ case 0 และจำทำไปเรื่อย ๆ จนกว่าจะเจอ break ซึ่งนั้นหมายความว่า ถ้าไม่มี break ก็จะทำงานไล่ไปทีละคำสั่งเรื่อย ๆ จนหมด โดยไม่สนใจว่า จะอยู่ใน case เดียวกันหรือไม่

8.10 REQUIRE

require จะแทนที่ตัวเองด้วยคำสั่งที่อยู่ใน file ที่กำหนดไว้ ซึ่ง require จะทำงานเพียงแค่ครั้งเดียว เมื่อโปรแกรมถูก load เท่านั้น

8.11 INCLUDE

จะคล้ายกับ require แต่จะต่างกันตรงที่ว่า include จะทำงานเมื่อถูกเรียก ไม่ใช่ทำงานเมื่อโปรแกรมถูก load ดังนั้น file ที่ include อาจจะเปลี่ยนไปเรื่อย ๆ ได้ (เช่น การใส่ include ไว้ใน loop ต่าง ๆ) แต่ว่า file ที่ require จะเป็นได้แค่ file เดียวที่กำหนดไว้ตอนแรก

8.12 FUNCTION

Function เป็นการรวมชุดคำสั่งต่าง ๆ ไว้ด้วยกัน เพื่อให้สะดวกต่อการเรียกใช้ โดยจะมีโครงสร้างดังนี้

```
Function foo ($arg_1, $arg_2, …., $arg_n) {
    Echo “Example function.\n”;
    Return $retval;
}
```

และมีส่วนประกอบดังนี้

- Function Name : เป็นชื่อที่จะถูกใช้ในการอ้างอิงถึง function นั้น
- Arguments ($arg) : Arguments คือค่าข้อมูลที่ส่งให้กับ function เพื่อใช้ในการประมวลผล
- Return Value : คือค่าผลลัพธ์ที่ได้จากการทำงาน ซึ่ง function จะคืนย้อนกลับออกมา

PHP Feature

นอกจากโครงสร้างภาษาที่ไม่ซับซ้อนแล้ว PHP ยังมีคุณสมบัติบางอย่างทีมีประโยชน์อย่างมากต่อการเขียน web programing อีกด้วย ในที่นี้จะยกมากล่าวถึงเพียง 3 หัวข้อที่นิยมใช้คือ

1. การจัดการข้อผิดพลาด (Error Handing)

ในกรณีที่เกิด error เกิดขึ้นเราสามารถที่จะดู message ของ error ได้โดยค่าจะอยู่ที่ global variable ชื่อ $php_errormsg

2. การจัดการรูปภาพ (Creating and manipulating images)

PHP สามารถสร้างและแก้ไขรูปภาพ นามสกุล .png, .jpg, .wbmp และ .xpm ได้ แต่ตอนติดตั้ง PHP จำเป็นจะต้องกำหนดให้ PHP ติดตั้ง GD Libray (library เกี่ยวกับการสร้างรูปภาพ) ลงไปด้วย ตัวอย่างการสร้างรูป .png เช่น

```
<?php
        Header("Content-type: image/png");
        $string=implode($argv,"");
        $im = imagecreatefromgif("images/button1.png");
        $orange = ImageColorAllocate($im, 220, 210, 60);
        px = (imagesx($im)-7.5*strlen($strubg))/2;
        ImageString($im, 3, $px, 9,$string, $orange);
        ImageGif($im);
        ImageDestroy($im);
?>
```

เป็นตัวอย่างการเรียก image ที่คล้าย <img src = "button.php?text"> แต่วิธีนี้จะยืดหยุ่นกว่าตรงที่เมื่อเราต้องการเปลี่ยน text ภายในปุ่มโดยไม่ต้องสร้างปุ่มใหม่

3. การจัดการถ่ายโอนแฟ้มข้อมูล (Handling file uploads)

PHP มีความสามารถในการ upload file ผ่าน web browser ได้ ซึ่งคุณสมบัตินี้สามารถ upload ทั้ง text file และ binary file ได้ ตัวอย่างของการเขียนฟอร์มเพื่อใช้ upload file

```
<HTML>
 <BODY>
    <FORM ENCTYPE = "multipart/form-data" ACTION="php-script" METHOD=POST>
        <INPUT TYPE = "hidden" name= "MAX_FILE_SIZE" value="1000">
        Send this file: <INPUT NAME = "userfile" TYPE = "file">
        <INPUT TYPE="submit" VALUE = "Send File">
     </FORM>
</BODY>
</HTML>
```

MAX_FILE_SIZE เป็นตัวแปรที่ใช้กำหนดขนาดของ file ที่ต้องการ upload (byte) ซึ่งจะกำหนดหรือไม่ก็ได้ userfile เป็นตัวแปรของ file ที่ต้องการ upload เมื่อผู้ใช้กด submit เพื่อส่งค่าต่าง ๆ ในฟอร์มมายัง php script แล้ว PHP

จะทำการกำหนดตัวแปรขึ้นมาเพื่อใช้จัดการกับ file upload โดยจากตัวอย่างข้างต้น ตัวแปร file ที่ส่งมาให้ชื่อ $userfile เพราะฉะนั้นตัวแปรที่ PHP ทำการสร้างขึ้นมาคือ

$userfile – ชื่อ file ชั่วคราวที่ถูก upload มาเก็บ ไว้ในเครื่อง server

$userfile_name – ชื่อ file ที่แท้จริงที่ได้จาก HTML Form

$userfile_size – ขนาดของ file ที่ถูก upload มีหน่วยเป็น byte

$userfile_type- ชนิดของ file ที่ถูก upload (ขึ้นกับว่า browser จะสามารถส่งข้อมูลส่วนนี้มาให้ server ได้หรือไม่) ตัวอย่างเช่น “image/gif” เป็นต้น

4. <u>การใช้งานสายอักบระ (Sring)</u>

สิ่งสำคัญประการหนึ่งในการเขียน web programming คือการสร้างและการใช้งาน string ซึ่งจะทำให้การเขียน php script มีประสิทธิภาพมากยิ่งขึ้น

4.1 <u>Quoted Strings</u>

สำหรับ PHP แล้ว string หมายถึงสิ่งที่อยู่ภายใต้เครื่องหมาย quotes (‘ หรือ “) ซึ่ง PHP จะตรวจสอบโดยการหาคู่ที่เหมือนกันของเครื่องหมาย เพราะฉะนั้นการสร้าง string จะต้องขึ้นต้นและปิดท้ายด้วยเครื่องหมาย quote ที่เหมือนกัน ตัวอย่างเช่น

```
‘I am a string in single quotes’
“I am a string in double quotes”
```

หากต้องการแทรกเครื่องหมาย quotes ใน string ต้องใช้เครื่องหมาย backslash (\) นำหน้า เพื่อบอก PHP ว่า quotes เป็นเพียงส่วนหนึ่งของ string ซึ่งการใช้เครื่องหมาย backslash ในการแทรกเครื่องหมาย quotes ลงในไป string นั้นทำให้ backslash กลายเป็นเครื่องหมายพิเศษ ซึ่งหากต้องการแทรกลงใน string ก็ต้องใช้เครื่องหมาย backslash นำหน้าเช่นกัน ตัวอย่างของการแทรกเครื่องหมาย quotes และ backslash คือ

```
‘I\m string in single quotes’            // การแทรกเครื่องหมาย single quotes
$file = “c:\windows\system.int”;        //ค่าที่เก็บอยู่ในตัวแปร $file คือ c:windowssystem.ini
$file = “c:\\windows\\syostem.ini”;     //ค่าที่เก็บอยู่ในตัวแปร $file คือ c:\windows\system.ini
```

นอกจากตัวอักขระทั้ง 2 ตัวนี้ ที่จำเป็นต้องใช้เครื่องหมาย backslash นำหน้าแล้ว ยังมีตัวอักขระกลุ่มหนึ่งเช่นกัน ซึ่งถือได้ว่าเป็นตัวอักขระพิเศษ (เหมือนกับภาษา c และ pert) ดังนี้

ตัวอักษรพิเศษ	ความหมาย
\n	Linefeed (LF or OxOA in ASCII)
\r	Carriage return (CR or 0x0D in ASCII)
\t	Horizontal tab (HT or 0x09 in ASCII)
\\	Backslash
\$	Dollar sign
\”	Double-quote
\[0-7]{1, 3}	The sequence of characters matching the regular expression is a character in octal notation
\x[0-9A-Fa-f]{1, 2}	The sequence of characters matching the regular expression is a character in

	hexadecimal notation

4.2 <u>การใช้ตัวแปร กับสายอักขระ (Using Variables in Strings)</u>

String ที่อยู่ในเครื่องหมาย double quoted และ single quotes นั้นถูก PHP จัดการด้วยวิธีการที่แตกต่างกัน นั่นคือ double quoted string ต้องถูกตีความในขณะที่ single quoted string จะไม่มีการตีความใด ๆ ทั้งสิ้น ดังที่ได้กล่าวมาแล้วข้างต้น ดังตัวอย่างต่อไปนี้

```
$foo = 2;
echo “foo is $foo”;     //ผลลัพธ์ที่ได้คือ : foo is  2
echo “foo is $foo”;     //ผลลัพธ์ที่ได้คือ : foo is $foo
echo “foo is $foo\n”;     //ผลลัพธ์ที่ได้คือ : foo is 2 (พร้อมกับการขึ้นบรรทัดใหม่)
echo “foo is $foo\n”;     //ผลลัพธ์ที่ได้คือ : foo is $foo\n
```

จากตัวอย่างจะเห็นว่า double quotes ควรถูกใช้งานในกรณีที่ต้องการกำหนดค่าของตัวแปรและอักขระพิเศษใน string ส่วน single quotes ควรถูกใช้งานในกรณีทั่วไปซึ่งการใช้ single quotes จะทำงานได้เร็วกว่า เพราะ PHP ไม่จำเป็นต้องประมวลผล string ที่อยู่ใน single quotes

Note : การใช้อักขระพิเศษใน single quotes สามารถใช้ได้เพียงแค่ \\ และ \’ เท่านั้น

แต่ในบางกรณี สำหรับตัวแปรที่ซับซ้อนมากก็ไม่สามารถเขียนลงใน double quotesได้ ตัวอย่าง

```
echo “value = $foo”;
echo “value = $a[$I]”;
```

สามารถทำได้

แต่ …

```
echo “value = $a[$I] [$j]”;
```

ไม่สามารถทำได้

และเพื่อทำเป็นการป้องกันปัญหาจากการใช้ตัวแปรใน string วิธีที่นิยมใช้คือใช้ string operator (.) ช่วยเชื่อม string กับตัวแปรเข้าด้วยกัน เช่น

```
echo ‘value = ‘ .$a[$I] [$j];
echo ‘value = ‘ .$this->var;
```

หรือใช้เครื่องหมายวงเล็บปีกกาล้อมรอบตัวแปรเหล่านี้ เช่น

```
$var = 3 ;
echo “value = {$var}”;  // Will print “value = 3”
echo “value = \{$var}”;   // Will print “value = {3}”
```

5. <u>ฟังก์ชันอ้างอิง (Function Reference)</u>

5.1 <u>String Function</u>

PHP มีฟังก์ชั่นที่ใช้จัดการกับ String อยู่เป็นจำนวนมาก ซึ่งสามารถหาอ่านได้จาก manual ของ PHP เพราะฉะนั้นในที่นี้จะยกมากล่าวเพียงบางฟังก์ชั่นที่เป็นที่นิยมใช้

ก่อนอื่นต้องทำความเข้าใจเบื้องต้นก่อนว่า sting ใน PHP ก็คือ zero indexed array of character ซึ่งหมายความว่า ตัวอักษรตัวแรกของ string อยู่ที่ตำแหน่งที่ 0 ของ array นั่นเอง ซึ่ง ทุก string function จะใช้ตำแหน่งที่ 0 เป็น index ของตัวอักษรตัวแรก

- AddSlashes

ใช้สำหรับเพิ่มอักษร "/" เอาไว้ข้างหน้าตัวอักษเพื่อที่จะได้เอาไว้อ้างอิงเวลาที่จะต้องการแยก String มีรูปแบบคือ

```
String addslashes(string str);
```

- Chop

ทำการตัดช่องว่างข้างหลังตัวอักษร มีรูปแบบคือ

```
String chop(string)
```

- Chr

แสดงค่าตัวอักษรตามรหัส ascii มีรูปแบบคือ

```
String chr(int ascii)
```

- echo

แสดงผลที่หน้าจอทุก ๆ parameter มีรูปแบบคือ

```
String chop(string agr1, …)
```

- explode

ใช้สำหรับ ทำการแบ่งสตริง ด้วย สตริงที่กำหนดให้ มีรูปแบบคือ

```
Array  explode(string separator, string string);
```

ตัวอย่างการใช้งาน

```
$pizza="piece1 piece2 piece3";
$piece=explode(" ",$pizza);
```

- implode

ใช้สำหรับเชื่อม array ให้เป็น string โดยที่มีตัวเชื่อมคือ string glue มีรูปแบบคือ

```
String implode(array piece, string glue);
```

ตัวอย่างการใช้งาน

```
$a=array(1, 2, 3, 4);
echo implode($a, ":");
```

จะได้ผลลัพธ์เป็น

1:2:3:4

- print

พิมพ์ output string คล้ายๆ กับ echo(string arg….); มีรูปแบบคือ

```
Print(string agr);
```

- strchr

ค้นหาอักษรหรือคำที่ต้องการ มีรูปแบบคือ

```
String strchr(string tofind, string toneed);
```

ตัวอย่างการใช้งาน

```
Echo (strchr(“I am a boy”, “am”));
```

- strcmp

ทำการเปรียบเทียบ sting ทั้งสอง มีรูปแบบคือ

```
Int strcmp(string str1, string str2);
```

ตัวอย่างการใช้งาน

0 – str1 = str2

1 – str1 > str2

-1 – str1 < str2

- strlen

แสดงความยาวของ string มีรูปแบบคือ

```
Int strlen (string str1)
```

- strrpos

แสดงตำแหน่งของตัวอักษรที่หาโดยที่ตัวอักษรตัวนั้นเป็นตัวสุดท้ายในประโยค มีรูปแบบคือ

```
String strrpos(string haystack, char needle);
```

ตัวอย่างการใช้งาน

```
Echo (strrpos(“I am a boy”,”a”));
```

จะได้ผลลัพธ์เป็น

5

- strpos

แสดงตำแหน่งของตัวอักษรที่หาได้เป็นตัวแรกในประโยค มีรูปแบบคือ

```
String strpos(string haystack, char needle);
```

ตัวอย่างการใช้งาน

```
Echo (strrrpos(“I am a boy”,”a”));
```

จะได้ผลผลลัพธ์เป็น

3

- strstr

ค้นหาคำที่ต้องการใน string มีรูปแบบคือ

```
String strst(string haystack, string needle)
```

ตัวอย่างการใช้งาน

echo (strstr(“I am a boy”,”am”));

จะได้ผลลัพธ์เป็น

am a boy

- strtolower

ทำการเปลี่ยนตัวอักษรให้เป็นตัวเล็กทั้งหมด มีรูปแบบคือ

```
string strtolower(string str1)
```

- strtoupper

ทำการเปลี่ยนตัวอักษรให้เป็นตัวใหญ่ทั้งหมด มีรูปแบบคือ

```
string strtoupper(string str1)
```

- substr

ทำการส่งค่าอักษรที่กำหนดให้ มีรูปแบบคือ

```
String substr (string string, int start, int length);
```

ตัวอย่างการใช้งาน

```
Echo (substr("I am a boy",2,4));
```

ได้ผลลัพธ์เป็น

am a

5.2 <u>Array Function</u>

- array

ใช้สำหรับสร้าง array มีรูปแบบคือ

```
Array array(….);
```

ดังตัวอย่างต่อไปนี้

```
<?
        $a=array(1,2,3,4,5);
        echo $a[1];  //ให้ทำการ แสดงค่า array ตำแหน่งที่ 1 จะได้ผลแสดงเป็น '2'  ออกมา
 ?>
```

```
<?
       $a=arrary(2=>"asdf") // เป็นการกำหนดค่าสตริง asdf ให้กับ array ตำแหน่งที่สอง
       echo $a[2];  // จะได้ผลเป็น asdf ออกมา
?>
```

- arsort

ใช้สำหรับเรียงลำดับ array แบบถอยหลัง มีรูปแบบคือ

```
Void arsort (array array);
```

ตัวอย่าง

```
<?
        $friuts = array(1=>"lemon",2=>"orange",3=>"banano",4=>"apple");
        arsort($fruits);
        for(reset($fruits); $key = key($fruits); next($fruits)){
           echo "fruits[$key] = ".$fruits[$key]."\n";
        }
```

```
?>
```

จะแสดงผลลัพธ์เป็น

fruits[2] = orange fruits[1] = lemon fruits[3] = banana fruits[4] = apple

- asort

ใช้สำหรับเรียงลำดับ array แบบเรียงลำดับ มีรูปแบบ คือ

```
Void asort (array array);
```

ดังตัวอย่าง

```
<?
    $fruits = array(1 => “emon”,2=>”orange”,3=>”banana”,4=>”apple”);
    arsort($fruits);
    for(reset($fruits); $key = key($fruits); next($fruits)) {
        echo “fruits[$key] = “.$fruits[$key].”\n”;
    }
?>
```

จะแสดงผลดังนี้

fruits[4] = apple fruits[3] = banana fruits[1] = lemon fruits[2] = orange

- current

ในแต่ละ array จะมีคล้าย ๆ กับ pointer เราเรียกว่า internal pointer ที่ใช้สำหรับชี้ค่าปัจจุบันที่ชี้อยู่ ซึ่ง current() จะทำหน้าที่แสดงค่าปัจจุบันออกมา มีรูปแบบคือ

```
Mixed current (array array): // mixed หมายถึง return ค่าได้หลายประเภท
```

ใช้ร่วมกันกับ ฟังก์ชั่น end(), prev(), next(), reset()

ตัวอย่าง

```
<?
    $fruits = arrayf(1=>”lemon”, 2=>”orange”,3=>”banano”,4=>”apple”);
    end($fruits);
    echo current($fruits);
?>
```

จะได้ผลเป็น apple ออกมา นั้นแสดงว่า internal pointer เลื่อนไปอยู่ที่ตำแหน่งสุดท้ายแล้ว

- end()

ใช้สำหรับ เลื่อน internal pointer ไปตำแหน่งสุดท้าย มีรูปแบบคือ

```
Mixed end (array array)
```

- next()

ใช้สำหรับ เลื่อน internal pointer ไปตำแหน่งถัดไปจากตำแหน่งปัจจุบัน มีรูปแบบคือ

```
Mixed next(array array)
```

- prev()

ใช้สำหรับ เลื่อน internal pointer ไปตำแหน่งก่อนหน้าตำแหน่งปัจจุบัน มีรูปแบบคือ

Mixed prev(array array)

- reset()

ใช้สำหรับ reset ค่า internal pointer ให้ไปอยู่ตำแหน่งแรก มีรูปแบบคือ

Reset(array array)

- sizeof()

ใช้สำหรับ แสดงจำนวน element ทั้งหมดที่มีอยู่ใน array มีรูปแบบคือ

Int zizeof(array array)

5.3 <u>Date/Time Function</u>

- checkdate

ใช้สำหรับตรวจสอบ วัน เดือน ปี ที่กำหนดว่าอยู่ในช่วงหรือไม่ จะอยู่ในช่วงก็จะ return ค่า 1 กลับ มีรูปแบบคือ

Int checkdate (int month, int date, int year)

- year is between 1900 and 32797 inclusive
- month is between 1 and 12 inclusive
- day is within the allowed number of days for the given month. Leap year are taken into consideration.

- Date

จะแสดงค่าวันที่เวลาตามที่กำหนดตาม formal ไป ส่วน timestamp จะเอาไว้สำหรับแสดงเวลาของสถานที่ต่าง ๆ บนโลก ถ้าไม่กำหนดจะแสดง locatime มีรูปแบบคือ

String date(string format, int timestamp)

Format string ใช้สำหรับแสดงเวลาแบบต่าง ๆ

U – seconds

Y – year, numeric, 4 digits

y - year, numeric, 2 digits

F – month, textual, long; i.e. “January”

M – month, textual, 3 letters; i.e. “Jan”

m - month, unmeric

z – day of the year, numeric

l (lowercase ‘L’) – day of the week, textual, long; i.e. “Friday”

D – day of the week, textual, 3 letters; i.e. “Fri”

w - day of the week, numeric, 1 digit

H – hour, numeric, 24 hour format

h - hour, numeric, 12 hour format

i - minutes, numeric

s - seconds, uumeric

A – “AM” or “PM”

a - “am”or”pm”

S – English ordinal suffix, textual, 2 characters; i.e. “th”, “nd”

ตัวอย่าง

```
<?
        Print(date(“I dS of F Y h:I:s A”));
?>
```

ผลที่ได้รับ Saturday 12 th February 2000 01:15:23 AM

- mktime

แสดงวินาที่นับตั้งแต่เวลาของ UNIX Epoch(January 1, 1970) จนถึงวันที่กำหนด มีรูปแบบคือ

```
Int mktime(int hour, int minute, int second, int month, int day, int year);
```

- time

แสดงวินาที่ที่นับตั้งแต่เวลาของ UNIX Epoch(January 1, 1970). มีรูปแบบคือ

```
Int time()
```

- Set_time_limit

กำหนดเวลาที่ script ทำงานอยู่เช่น set_time_limit(30) หลังจากนั้น scripts ก็จะ fatal error มีรูปแบบคือ

```
Void set_time_limit (int seconds);
```

5.4 <u>Filesystem Function</u>

- file_exists

ใช้ตรวจสอบไฟล์ที่ต้องทราบว่ามีอยู่แล้วไม่ ซึ่งหาก filename มีอยู่แล้วจะ return ค่าเป็นจริง หากไม่มี return ค่าเป็นเท็จ มีรูปแบบคือ

```
Int file_exists(string filename);
```

- fileatime

แสดงเวลาการแข้ไข file ครั้งสุดท้าย มีรูปแบบ คือ

```
Int fileatime(string filename);
```

- filesize

แสดงขนาดของ file มีหน่วยเป็น byte

```
Int filesize (string filename)
```

- file

อ่านข้อความทั้งหมดใน file มาเก็บไว้ใน array ซึ่งแต่ละสมาชิกใน array จะถูกแทนด้วยข้อความแต่ละบรรทัดใน file มีรูปแบบคือ

```
Array file(string filename)
```

- fopen

เปิด File หรือ URL มีรูปแบบ คือ

```
int fopen(string filename, string mode)
```

file จะภูกเปิดจาก filesystem และจะมีการ retrun file pointer กลับมา หากไม่สามารถเปิด file ได้ function จะ return เท็จ

mode ทั้งหมด มีดังนี้

- 'r' - เปิดเพื่ออ่านอย่างเดียว, file pointer ชี้ไปยังตำแหน่งเริ่มต้นของ file
- 'r+' - เปิดเพื่ออ่านและเขียน, file pointer ชี้ไปยังตำแหน่งเริ่มต้นของ file
- 'w' - เปิดเพื่อเขียนอย่างเดียว, file pointer ชี้ไปยังตำแหน่งเริ่มต้นของ file และทำการลบข้อมูลใน file ทั้งหมด หาก file ไม่มีอยู่จริง จะพยายามสร้าง file ใหม่ขึ้นมา
- 'w+' - เปิดเพื่ออ่านและเขียน, file pointer ชี้ไปยังตำแหน่งเริ่มต้นของ file และทำการลบข้อมูลใน file ทั้งหมด หาก file ไม่มีอยู่จริง จะพยายามสร้าง file ใหม่ขึ้นมา
- 'a' - เปิดเพื่อเขียนอย่างเดียว, file pointer ชี้ไปยังตำแหน่งสุดท้ายของ file (EOF:end of file) หาก file ไม่มีอยู่จริง จะพยายามสร้าง file ใหม่ขึ้นมา
- 'a+' - เปิดเพื่ออ่านและเขียน, file pointer ชี้ไปยังตำแหน่งสุดท้ายของ file (EOF:end of file) หาก file ใหม่ขึ้นมา

ตัวอย่างเช่น

```
$fp = fopen("/home/rasmus/file.txt","r");
$fp = fopen("/http://www.php.net/","r");
 $fp = fopen("/ftp://user:password@example.com/","w");
```

- fclose

ปิด file pointer ที่เปิดอยู่ ซึ่งจะ return ค่าเป็นจริงหากสามารถเปิด file pointer ได้

```
int fclose(int fp)
```

- fgetc

retrun ตัวอักษรครั้งละ 1 ตัว จากตำแหน่งที่ file ชี้อยู่ หาก file pointer ชี้ที่ EOF จะ reutrn ค่าเป็นเท็จ

```
string fgetc(int fp)
```

- fgets

return ตัวอักษรครั้งละ length-1 bytes ที่อ่านจากตำแหน่งที่ file pointer ชี้อยู่ ซึ่งการอ่านตัวอักษรอาจสิ้นสุดลงหาก PHP สามารถอ่านได้ length-1 หรือ file pointer ชี้ไปยัง new file หรือ EOF รูปแบบคือ

```
string fgets(ing fp, int length)
```

ตัวอย่างเช่น

```
$fd = fopen("/tmp/inputfile.txt","r");
while ($buffer = fgets($fd, 4096)) {
echo $buffer;
}
fclose($fd);
```

- fread

อ่านข้อความใน file ซึ่งการอ่านจะสิ้นสุดหาก PHP สามารถอ่านได้ตาม length (ความยาว) ที่กำหนด หรือ file pointer ชี้ไปยัง EOF มีรูปแบบคือ

```
string fread(int fp, string length)
```

ตัวอย่างเช่น

```
// get contents of a file into a string
$filename = "/usr/local/something.txt"
$fd = fopen($filename, "r");
$contents = fread($fd, filesize($filename));
fclose($fd);
```

- fwrite

เขียนข้อความลง file มีรูปแบบคือ

```
int fwrite(int fp, string string,int[length])
```

5.5 MySQL functions

function ในภาษา PHP ส่วนใหญ่ที่เกี่ยวข้องกับ MySQL ก็จะเป็นเรื่องการจัดการข้อมูลเป็นสำคัญ ซึ่งใน function ทั้งหมด จะใช้ Link Identifier เป็นตัวบอกว่า กำลังติดต่อกับ Database ตัวในอยู่ สำหรับ รายละเอียดของแต่ละ function นั้นจะไล่ไปตามลำดับอักษร เพื่อให้สามารถค้นหาได้ง่าย ดังนี้

- mysql_close

ใช้เมื่อต้องการยกเลิกการติดต่อกับ MySQL โดยจะมีรูปแบบดังนี้

```
int mysql_close(int [link_identifier]);
```

โดย function นี้ต้องการ argument หนึ่งตังคือ link Identifier ซึ่งจะได้จากคำสั่ง mysql_connect และ ถ้าสามารถยกเลิกได้สำเร็จ function นี้จะ return ค่า TURE ออก ถ้ายกเลิกไม่สำเร็จ จะ return ค่า FALSE

- mysql_connect

ใช้เมื่อต้องการเชื่อมต่อสู่ MySQL โดยจะมีรูปแบบคำสั่งดังนี้

```
int mysql_connect(string [hostname] [:port], string [username] ,string [password]);
```

function นี้ต้องการ argument 3 ตัว ที่เกี่ยวข้องกับการเชื่อมต่อกับ MySQL คือ hostname คือ ชื่อเครื่องที่มี MySQL ทำงานอยู่ (MySQL Server) ซึ่งเราต้องการเชื่อมต่อด้วย port คือหมายเลขของ port ที่ให้บริการ MySQL บนเครื่อง MySQL Server ซึ่งจะใส่หรือไม่ใส่ก็ได้ ถ้าไม่ใส่ ก็จะใช้ค่ามาตรฐาน

username คือ ชื่อผู้ใช้ที่มีสิทธิ์ใช้ MySQL

password คือ รหัสผ่านสำหรับผู้ใช้คนนั้น

ถ้าทำการเชื่อมต่อได้ function นี้จะ return ค่า link Identifier ออกมา (เพื่อนำไปใช้ในการจัดการกับฐานข้อมูล MySQL ต่อไป) แต่ถ้าเชื่อมต่อไม่ได้ ก็จะ return ค่า FALSE ออกมา

ตัวอย่างเช่น

```
// create connection
$connection = mysql_connect("servername","username","password");


// test connection
if (!$connection) {
        echo "Couldn't make a connection!";
        exit;
}
```

- mysql_create-db

ใช้เมื่อต้องการสร้าง Database ใหม่ขึ้นมา มีรูปแบบคือ

```
int mysql_create_db(string database name, int [link_identifier]);
```

โดยต้องการ argument คือ ชื่อ Database ที่ต้องงการสร้างขึ้นมาใหม่ และ Link Identifier ของ MySQL ที่ต้องการสร้าง database

- mysql_db_query

ใช้เมื่อต้องการส่ง SQL message ไปยัง MySQL เพื่อ query ข้อมูล มีรูปแบบคือ

```
int mysql_db_query(string database, string query, int [link_identifier]);
```

มี argument คือ ชื่อ Database ที่ต้องการ query, SQL message ที่ใช้ในการ query และ Link Identifier ของ MySQL ที่มี Database นั้นอยู่ ในกรณีที่ query เป็นผลสำเร็จก็จะ Return ค่าเป็น result Identifier ที่ใช้ไปยัง result set ที่เก็บผลการทำงาน ออกมา แต่ถ้า query ไม่สำเร็จ ก็จะ return FALSE ออกมา

- mysql_drop_db

ใช้ในกรณีที่ต้องการลบ Database ทิ้งจาก MySQL มีรูปแบบคือ

```
int mysql_drop_db(string database_name, int [link_identifier]);
```

โดยมี argument คือ ขื่อ Database ที่ต้องการลบ และ Link Identifier ของ MySQL ที่ต้องการติดต่อได้ ถ้าลบได้สำเร็จ ก็จะ Retrun TRUE ออกมา ถ้าลบไม่สำเร็จ ก็จะ return FALSE ออกมา

- mysql_errno

แสดง error number ของคำสั่งก่อนหน้าที่ มีรูปแบบคือ

```
int mysql_errno(int [link_identifier]);
```

- musql_error

แสดง error message ของ คำสั่งก่อนหน้านี้ มีรูปแบบคือ

```
string mysql_error(int [link_identifier]);
```

- mysql_fetch_array

อ่านข้อมูลผลลัพธ์ในรูปของ Associative Array มีรูปแบบคือ

```
array myssql_fetch_array(int result, int [result_type]);
```

ต้องการ argument คือ ชื่อ result identifier และ ค่า result type ในกรณีที่มีผลลัพธ์ที่ต้องการ ก็จะ return ค่า array ของผลลัพธ์ออกมา ซึ่งสามารถดึงค่าข้อมูลออกมาได้ โดยอาศัย field name นั่นเอง ในกรณีที่เกิดปัญหา ก็จะ return FALSE ออกมาแทน

- mysql_fetch_length

ใช้ในการดูขนาดของข้อมูลที่ใหญ่ที่สุดของผลลัพธ์แต่ละตัว ใน result set มีรูปแบบคือ

```
array mysql_fetch_length(int result);
```

โดยจะ return ค่า array ของ ขนาดข้อมูลในแต่ละ field ของ row สุดท้ายที่ ถูก fetch ออกมา หรือไม่ ค่า FALSE ถ้าเกิดปัญหาขึ้น

- mysql_fetch_row

ดึงข้อมูลออกมาหนึ่ง row จาก result set ในรูปของ array มีรูปแบบคือ

```
array mysql_fetch_row(int result);
```

ต้องการ argument คือ result identifier และ return ค่า array ที่เก็บข้อมูลของ row แรก (ถ้าเรียก mysqll_fetch_row อีกครั้ง ก็จะเป็นการเรียก row ถัดๆ ไปเรื่อยๆ) ออกมา หรือ retrun FALE ถ้าเกิดปัญหาขึ้น

array ที่เก็บข้อมูลนั้น เป็น Scalar Array ซึ่งจะอ้างอิงข้อมูลเรียงลำดับตาม field และมี Index เริ่มต้นเป็น 0

- mysql_field_name

อ่านชื่อของ field จาก result set มีรูปแบบคือ

```
string mysql_field_name(int result, int field_index);
```

เมื่อ argument คือ result identifier และ field index ที่ต้องการอ่านชื่อออกมา

- mysql_field_table

อ่านชื่อ table จาก result set ที่ field ที่กำหนด อยู่ด้านในมีรูปแบบคือ

```
string mysql_field_table(int result, int field_offset);
```

เมื่อ argument คือ result identifier และ field offset ของ table ที่ต้องการอ่านชื่อออกมา

- mysql_field_type

อ่านชนิดของ field จาก result set ตามที่ต้องการ มีรูปแบบคือ

```
string mysql_field_type(int result, int field_offset);
```

ต้องการ argument คือ result identifier และ field offset ที่ต้องการหาชนิด

- mysql_field_len

อ่านความยาของ field จาก result set ตามที่ต้องการ มีรูปแบบ คือ

```
string mysql_field_len(int result, int field_offset );
```

ต้องการ argument คือ result identifier และ field offset ที่ต้องการหาความยาว

- mysql_ free_result

ลบ result set ทิ้ง มีรูปแบบคือ

```
int mysql_free_result(int result);
```

ต้องการ argument คือ result identifier ของ result set ที่ต้องการลบทิ้ง ซึ่งการลบ result set ที่ไม่ต้องการใช้ทิ้งนั้น จะช่วยประหยัด memory ของระบบได้มาก

- mysql_list_fileds

แสดงรายชื่อ field ของ Table ใน Database ที่ต้องการออกมา มีรูปแบบคือ

```
int mysql_list_fields(string database_name, string table_name, int[link_identifer]);
```

ต้องการ argument คือ ชื่อ Database และ ชื่อ Table ที่ต้องการแสดง Field ออกมา โดยจะ return ค่า result identifier ของ result set ออกมา ซึ่งต้องเอาไปใช้กับ function ในกลุ่ม mysql_field_*() ทั้งหลาย

- mysql_list_dbs

แสดงรายการของ Database ใน MySQL ที่ติดต่อด้วย มีรูปแบบคือ

```
int mysql_list_dbs(int [link_identifier]);
```

มี argrment คือ Link Identifier และจะ return ค่า result identifier ออกมา และต้องใช้ function mysql_tablename() เพื่ออ่านค่าใน result set นี้ออกมา

- mysql_listK_tables

แสดงรายการของ Tables ของ Database ใน MySQL ที่ติดต่อด้วย มีรูปแบบคือ

```
int mysql_list_tables(string database, int [link_identifier]);
```

มี argument คือ Link Identifier และ Database name และ จะ return ค่า result identifier ออกมา และต้องใช้ function mysql_tablename() เพื่ออ่านค่าใน result set นี้ออกมา

- mysql_num_fields

อ่านจำนวน fields จาก result set มีรูปแบบคือ

```
int mysql_num_fields(int result);
```

โดย function นี้ จะ return ค่า จำนวน fields ใน result set ที่ชี้โดย result identifier ที่กำหนดให้ออกมา

- mysql_num_rows

อ่านจำนวน rows จาก result set มีรูปแบบคือ

```
int mysql_num_rows(int result);
```

โดย function นี้จะ return ค่าจำนวน rows ใน result set ที่ชี้โดย result identifier ที่กำหนดให้ออกมา

- mysql_query

ส่ง SQL Query Message ไปยัง MySQL มีรูปแบบคือ

```
int mysql_query(string query, int [link_identifier]);
```

โดย function นี้จะ return TRUE เมื่อ Query เป็นผลสำเร็จ สำหรับกรณีที่ Query Message เป็นแบบ UPDATE, INSERT หรือ DELETE หรือ Result Set สำหรับ Query Message แบบ SELECT

- mysql_result

อ่านข้อมูลใน field ที่ต้องการ ออกจาก result set มีรูปแบบคือ

```
int mysql_result(int result, int row, mixed field);
```

ต้องการ qrgument คือ result identifier, row และ field offset หรือว่า field name หรือว่า field table.field name หรือว่า aliased ของ field นั้น ก็ได้

- mysql_select_db

เลือก Databse ใน MySQL มีรูปแบบคือ

```
int mysql_select_db(string database_name, int [link_identifier]);
```

โดย argument คือ ชื่อ Database ที่ต้องการเลือก และ Link Identifier โดยจะ return TRUE ถ้าเลือกได้ สำเร็จ และ FALSE เมื่อเกิด Error

ตัวอย่าง การใช้ PHP ติดต่อกับฐานข้อมูลของ MySQL
วิธีที่1

```
<?php

// create connection
$connection = mysql_connect("servername","username","password");

// test connection
if (!$connectiion) {
   echo "Couldn't make a connection!";
   exit;
}

// select database named "myDB"
$db = mysql_select_db("myDB", $connection);

// test selection
if (!$db) {
   echo "couldn't select database!";
   exit;
}

// create SQL statement
$sql = "SELECT COFFEE_NAME, ROAST_TYPE, QUANTITY
          FROM COOEE_INVENTORY
          ORDER BY QUANTITY DESC";

// execute SQL query and get result
$sql_result = mysql_query($sql, $connection);

// start results formatiion
echo "<TABLE BORDER=1>";
echo "<TR><TH>Coffee Naem</TH><TH>Roast Type</TH><TH>Quantity</TH>";
```

```
// format results by row
while ($row = mysql_fetch_array($sql_result)) {
$coffee_name = $row["COFFEE_NAME"];
$roast_type = $row["ROAST_TYPE"];
$quantity = $row["QUANTITY"];
echo "<TR><TD>$coffee_name</TD><TD>$roast_type</TD><TD>$quantity</TD></TD>";
}

echo "</TABLE>";

// free resources and close connectiion
mysql_free_result($sql_result);
mysql_close($connection);
?>
```

วิธีที่ 2

```
<?
include("mydb.inc");
mysql_connect(localhost,$user,$password);

$databse = "mydb";
@mysql_select_db("$database") or die("Unable to select database");

/*Search database for name*/
$table="contact";
$query="select * from $table group by name";
$result=mysql_query($query);
mysql_close();

/*Display Results*/
$num=mysql_numrows($result);

$i=0;
while($i < $num) {
```

```
    $name=mysql_result($result,$i,"name");
    $email=mysql_result($result,$i,"email");
    $ext=mysql_result($result,$i,"extension");
    $nick=mysql_result($result,$i,"nick");
    $id=mysql_result($result,$i,"id");

     //Alternate row colors in our table
     print ($i % 2) ? "<tr bgcolor=\"D7DEFF\">" : "<tr bgcolor =\"C2CCFF\">";

     print "<td align=middle><b>$name</b></td><td align=middle><$email</b></td>";
     print "<td align=middle><b>$ext</b></td><td align=middle><b>$nick</b></td>";
     print "<td align=middle><a href='update_db.phtml?id=$id'><b>Update</b> </a></td></tr>";

     ++$i;
}
?>
```

****