

ทฤษฎีบทที่ 6 การใช้ PHP จัดการกับข้อมูล

6.1 การทำงานกับข้อความ (String)

สตริง (String) คือกลุ่มของอักขระทั้งที่เป็นตัวอักษร ตัวเลข และอักขระพิเศษ ซึ่งสามารถสร้างขึ้นได้ 3 วิธี

- ใช้สัญลักษณ์ ' (single quote)

ตัวอย่าง

ผลลัพธ์

<pre><?php print 'ทดสอบการสร้างสตริง'; ?></pre>	ทดสอบการสร้างสตริง
---	--------------------

ตัวอย่าง

ผลลัพธ์

<pre><?php print 'ทดสอบการสร้าง "สตริง"'; ?></pre>	ทดสอบการสร้าง "สตริง"
--	-----------------------

- ใช้สัญลักษณ์ " (double quote)

ตัวอย่าง

ผลลัพธ์

<pre><?php print "ทดสอบการสร้าง สตริง"; ?></pre>	ทดสอบการสร้างสตริง
--	--------------------

6.2 ฟังก์ชันที่ใช้จัดการกับข้อความ (String Functions)

ใน PHP มีฟังก์ชันที่ใช้จัดการกับข้อความเป็นจำนวนมาก โดยในบทนี้จะขออธิบายถึงฟังก์ชันที่ต้องใช้บ่อยและฟังก์ชันที่ควรทราบ เพื่อเพิ่มประสิทธิภาพของโปรแกรมหรือช่วยในการเขียนโปรแกรมทำได้ง่ายขึ้น

การแสดงผลข้อความบนจอภาพ (คำสั่ง echo, print)

คำสั่งหรือฟังก์ชันทั้งสองนี้ใช้ในการแสดงผลข้อความบนหน้าเว็บเพจ หรือใช้ส่งแท็กในภาษา HTML ไปยังเบราว์เซอร์ ซึ่งการใช้ echo และ print นี้สามารถใช้หรือไม่ใส่วงเล็บก็ได้

ตัวอย่างที่ 6.1

การใช้งาน echo และ print การใช้วงเล็บหรือไม่ใช้ ไม่มีผลต่อการแสดงผล

<pre>1 <?php 2 echo ("HTML
"); 3 echo "JavaScript
"; 4 print ("ASP
"); 5 print "PHP"; 6 ?></pre>	ผลลัพธ์ HTML JavaScript ASP PHP
---	---

สิ่งที่แตกต่างกันระหว่าง echo กับ print คือ echo ไม่สามารถใช้ได้ในฟังก์ชันตัวแปร (Variable function) แต่ print สามารถใช้ได้ดังตัวอย่างที่ 6.2

ตัวอย่างที่ 6.2

เปรียบเทียบให้เห็นความแตกต่างระหว่างคำสั่ง `echo` และ `print`

```

1 <?php
2 $a = 8;
3 function TestSet($pa) {
4     if (isset($pa)) {
5         print "\$a = $pa";
6     } else {
7         print "\$a was not set value.";
8     }
9 }
10 $xSet = "TestSet";
11 $xSet (print($a));
12 ?>

```

ผลลัพธ์

8\$a = 1

แต่ถ้าหากเปลี่ยนคำสั่ง `print` ที่บรรทัดรองสุดท้ายเป็น `echo` จะเกิดข้อผิดพลาด ดังนี้

Parse error: parse error, unexpected T_ECHO, expecting ')' in
c:\appserv\www\lesson6_2550\ex6_2.php on line 11

ฟังก์ชันตัวแปร (Variable functions) หมายถึงตัวแปรที่มีค่าเป็นชื่อฟังก์ชันใดฟังก์ชันหนึ่ง แล้วถูกเติมเครื่องหมายวงเล็บเปิดและปิดต่อท้าย (อาจมีอาร์กิวเมนต์หรือไม่ขึ้นอยู่กับแต่ละฟังก์ชัน) ตัวแปรนั้นจะสามารถทำหน้าที่ได้เหมือนกับฟังก์ชันที่มันเก็บชื่อเอาไว้ ดังตัวอย่างข้างบนที่กำหนดให้ตัวแปร `$xSet` มีค่าเป็นชื่อของฟังก์ชัน `TestSet()` เราจึงสามารถเรียกใช้ฟังก์ชัน `TestSet()` ผ่านทางตัวแปร `$xSet` ได้

การตัด แทนที่ และแทรกตัวอักษรในสตริง (ฟังก์ชัน `substr()`, `substr_replace()`)

ฟังก์ชัน `substr()` เป็นฟังก์ชันที่มีการใช้งานบ่อยฟังก์ชันหนึ่ง ใช้สำหรับตัดตัวอักษรหรือข้อความออกมาจากข้อความเดิม มีรูปแบบการใช้งาน ดังนี้

รูปแบบ

string substr (string *string*, int *start* [, int *length*])

string ข้อความที่ต้องการนำมาตัดตัวอักษร

start ตำแหน่งเริ่มต้นของตัวอักษรที่ต้องการตัด มีค่าเป็นบวกหรือลบก็ได้ โดยตัวอักษรตัวแรกเริ่มที่ค่า 0

length จำนวนตัวอักษรที่ต้องการตัด มีค่าเป็นบวกหรือลบก็ได้ และจะระบุหรือไม่ก็ได้ หากระบุไว้เป็นค่าบวก จะทำการตัดตั้งแต่ตำแหน่ง start ไปเป็นจำนวนเท่ากับ length แต่หากไม่ระบุจะทำการตัดตั้งแต่ตำแหน่ง start ไปจนสุดข้อความ

ตัวอย่างที่ 6.3	ตัวอย่างการตัดตัวอักษรออกมาจากข้อความด้วยฟังก์ชัน <code>substr()</code>
<pre> 1 <?php 2 \$oldstr = "Puff, the magic dragon, lived by the sea."; 3 print substr(\$oldstr, 6, 16); 4 ?> </pre>	ผลลัพธ์ the magic dragon

กรณีที่ค่าเริ่มต้น (start) เป็นลบ จะทำการตัดจากตัวอักษรตัวสุดท้ายเข้ามาเป็นจำนวนเท่ากับที่ระบุไว้ใน start

ตัวอย่างที่ 6.4	ตัวอย่างการตัดตัวอักษรออกมาจากด้านท้ายข้อความด้วยฟังก์ชัน <code>substr()</code>
<pre> 1 <?php 2 \$oldstr = "Puff, the magic dragon, lived by the sea."; 3 \$newstr = substr(\$oldstr, -4); 4 print \$newstr; 5 ?> </pre>	ผลลัพธ์ sea.

กรณีที่ค่าความยาว (length) เป็นลบ จะทำการตัดตัวอักษรทิ้ง นับจากตัวอักษรสุดท้ายเข้ามาเป็นจำนวนเท่ากับที่ระบุไว้ใน length ตัวอย่างเช่น

ตัวอย่างที่ 6.5	การตัดตัวอักษรออกมาจากข้อความด้วยฟังก์ชัน <code>substr()</code> กรณีความยาวเป็นลบ
<pre> 1 <?php 2 \$oldstr = "Puff, the magic dragon, lived by the sea."; 3 \$newstr = substr(\$oldstr, 0, -4); 4 print \$newstr; 5 ?> </pre>	ผลลัพธ์ Puff, the magic dragon, lived by the

หากค่าตำแหน่งเริ่มต้น (start) เกินกว่าความยาวของข้อความลบด้วย 1 (ตัวอักษรตัวแรกนับเป็นตำแหน่งที่ 0) ฟังก์ชัน `substr()` จะให้ค่าเป็นเท็จ (false) ตัวอย่างเช่น

ตัวอย่างที่ 6.6	การใช้ฟังก์ชัน <code>substr()</code> กรณีระบุตำแหน่งเริ่มต้นเกินกว่าความยาวของข้อความ
<pre> 1 <?php 2 \$oldstr = "Puff, the magic dragon, lived by the sea."; 3 \$newstr = substr(\$oldstr, 41, 4); 4 if (\$newstr) { 5 print \$newstr; 6 } else { 7 print "The message is shorter than the start."; 8 } 9 ?> </pre>	ผลลัพธ์ The message is shorter than the start.

นอกจากการตัดข้อความด้วยฟังก์ชัน `substr()` แล้ว ในกรณีที่ต้องการแทรกหรือแทนที่ตัวอักษร เราสามารถใช้ฟังก์ชัน `substr_replace()` ได้ซึ่งมีรูปแบบการใช้งานดังนี้

รูปแบบ

string `substr_replace` (string *string*, string *replacement*, int *start* [, int *length*])

string ข้อความที่ต้องการนำตัวอักษรหรือข้อความอื่นมาแทรกหรือแทนที่ตัวอักษร
replacement ข้อความหรือตัวอักษรที่นำมาแทรกหรือแทนที่

start	ตำแหน่งเริ่มต้นของตัวอักษรที่ต้องการแทรกหรือแทนที่ มีค่าเป็นบวกหรือลบก็ได้ โดยตัวอักษรตัวแรกเริ่มที่ค่า 0
length	จำนวนตัวอักษรที่ต้องการแทนที่ มีค่าเป็นบวกหรือลบก็ได้ และจะระบุหรือไม่ก็ได้ หากระบุไว้เป็นค่าบวก จะทำการแทนที่ตั้งแต่ตำแหน่ง start ไปเป็นจำนวนเท่ากับ length แต่หากไม่ระบุ ตัวอักษรตั้งแต่ตำแหน่ง start ไปจนสุดข้อความจะถูกแทนที่ด้วยค่าใน replacement

ตัวอย่างที่ 6.7

การเพิ่มเลขที่เอกสารจากค่าเดิมขึ้นไปอีกหนึ่ง โดยจะต้องทราบตำแหน่งเริ่มต้นและความยาวของตัวอักษรที่ต้องการแทนที่

<pre> 1 <?php 2 function IncreaseDocNo(\$pDocNo) { 3 \$runno = substr(\$pDocNo, 11, 2) +1; 4 if ((strlen(\$runno) >= 1) & (strlen(\$runno) <= 2)) { 5 \$runno = (strlen(\$runno) == 1) ? ("0" . \$runno) : \$runno; 6 \$newdocno = substr_replace(\$pDocNo, \$runno, 11, 2); 7 return \$newdocno; 8 } else { 9 return "Running Error!"; 10 } 11 } 12 print "New document no. : " . IncreaseDocNo("045/0061AC-02"); 13 ?> </pre>	ผลลัพธ์ New document no. : 045/0061AC-03
--	--

การแทนที่ข้อความโดยไม่ระบุจำนวนตัวอักษรที่จะถูกแทนที่ (length) จะมีผลให้ข้อความตั้งแต่ตัวอักษรที่เริ่มถูกแทนที่ไปจนสุดข้อความหายไปทั้งหมด ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 6.8

การแทนที่ข้อความโดยไม่ระบุจำนวนตัวอักษรที่ถูกแทนที่

<pre> 1 <?php 2 \$str = "I'll do my crying in the rain"; 3 print substr_replace(\$str, "sleeping", 11); 4 ?> </pre>	ผลลัพธ์ I'll do my sleeping
---	---

กรณีที่ length เป็นค่าลบ ข้อความจะถูกแทนที่ตั้งแต่ตำแหน่งเริ่มต้นที่ระบุไว้ไปจนถึงตำแหน่งที่นับจากท้ายข้อความเข้ามาเป็นจำนวนเท่ากับ length และกรณีที่ start มีค่าเป็นลบ ข้อความจะถูกแทนที่โดยเริ่มนับจากด้านท้ายข้อความเข้ามาเป็นจำนวนเท่ากับ start ดังตัวอย่าง

ตัวอย่างที่ 6.9

การแทนที่ข้อความเมื่อ length มีค่าเป็นลบ

<pre> 1 <?php 2 \$str = "I'll do my crying in the rain"; 3 print substr_replace(\$str, "sleeping", 11, -9); 4 print "
"; 5 print substr_replace(\$str, "sleeping", -18, -9); 6 ?> </pre>	ผลลัพธ์ I'll do my sleeping the rain I'll do my sleeping the rain
---	--

กรณีที่ length มีค่าเป็นศูนย์ จะเป็นการแทรกข้อความ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 6.10	การแทนที่ข้อความเมื่อ length มีค่าเป็นศูนย์
<pre>1 <?php 2 \$str = "I'll do my crying in the rain"; 3 print substr_replace(\$str, "falling ", -4, 0); 4 ?></pre>	ผลลัพธ์ I'll do my crying in the falling rain

การแทนที่คำหรือตัวอักษรในสตริง (ฟังก์ชัน str_replace())

ในกรณีที่ต้องการแทนที่ตัวอักษรหรือคำในข้อความด้วยตัวอักษรหรือคำอื่น เราสามารถใช้ฟังก์ชัน str_replace() ซึ่งมีรูปแบบดังนี้

รูปแบบ

mixed str_replace (mixed search, mixed replace, mixed subject [, int &count])

search คำที่ต้องการให้ถูกแทนที่

replace คำที่จะนำมาแทนที่

subject ประโยคหรือข้อความ

count ตัวแปรสำหรับรับค่าจำนวนคำหรือส่วนของข้อความที่ถูกแทนที่ ซึ่งจะถูกส่งผ่านโดยการอ้างอิง (pass by reference) ซึ่งจะต้องมีเครื่องหมาย & อยู่หน้าตัวแปร

ตัวอย่างที่ 6.11	การแทนที่คำในข้อความหลักด้วยฟังก์ชัน str_replace()
<pre>1 <?php 2 \$oldstr = "Puff, the magic dragon, lived by the sea."; 3 \$newstr = str_replace("dragon", "lizard", \$oldstr); 4 print \$newstr; 5 ?></pre>	ผลลัพธ์ Puff, the magic lizard, lived by the sea.

ฟังก์ชัน str_replace() สามารถคืนค่าให้ได้ทั้งสตริงและอาร์เรย์ ส่วนอาร์กิวเมนต์ search, replace และ subject ก็สามารรถเป็นได้ทั้งสตริงและอาร์เรย์ หากต้องการให้มีการแทนที่คำมากกว่า 1 คำ เราสามารถกำหนดให้ search และ replace เป็นตัวแปรอาร์เรย์ได้ และหากมีข้อความมากกว่า 1 ข้อความที่ต้องการให้มีการแทนที่คำ เราก็สามารถกำหนดให้ subject เป็นตัวแปรอาร์เรย์ของข้อความหลาย ๆ ข้อความได้ ซึ่งจะทำให้ฟังก์ชัน str_replace() คืนค่ากลับมาเป็นอาร์เรย์เช่นกัน

ตัวอย่างที่ 6.12	การแทนที่คำในข้อความหลักในแบบอาร์เรย์ด้วยฟังก์ชัน str_replace()
<pre>1 <?php 2 \$ar_text = array("Puff, the magic dragon, lived by the sea.", 3 "Beyond the sea"); 4 \$ar_word1 = array("dragon", "sea"); 5 \$ar_word2 = array("lizard", "sand"); 6 \$ar_newstr = str_replace(\$ar_word1, \$ar_word2, \$ar_text); 7 8 foreach (\$ar_newstr as \$newstr) { 9 print "\$newstr
"; 10 } 11 ?></pre>	ผลลัพธ์ Puff, the magic lizard, lived by the sand. Beyond the sand

การหาตำแหน่งของคำหรือตัวอักษร (ฟังก์ชัน strpos())

ภาษา PHP มีฟังก์ชันที่ช่วยให้การหาตำแหน่งของตัวอักษรหรือคำในข้อความทำได้ง่ายขึ้น ด้วยฟังก์ชัน strpos() เพียงแต่ใส่อาร์กิวเมนต์ที่เป็นข้อความที่ต้องการตรวจสอบ และตัวอักษรหรือคำที่ต้องการค้นหาลงไปเท่านั้น มีรูปแบบการใช้งานดังนี้

รูปแบบ

int strpos (string haystack, string needle [, int offset])

haystack ข้อความที่ต้องการตรวจสอบ

needle ข้อความหรือตัวอักษรที่ต้องการค้นหา

offset ตำแหน่งที่ให้เริ่มค้นหา (อักขระตัวแรกนับเป็น 0) เป็น optional parameter คือจะระบุหรือไม่ก็ได้

ผลลัพธ์ที่ได้จากฟังก์ชัน strpos() จะเป็นตำแหน่งแรกที่พบข้อความหรือตัวอักษรที่ต้องการค้นหา โดยตัวอักขระตัวแรกของ

ข้อความจะเป็นตำแหน่ง 0

ตัวอย่างที่ 6.13

การหาชื่อผู้ใช้ และชื่อโดเมนจากอีเมลแอดเดรส

```
1 <?php
2 $emailaddr = "somsakkee@yahoo.com";
3 $name = substr($emailaddr, 0, strpos($emailaddr, "@"));
4 $domain = substr($emailaddr, strpos($emailaddr, "@") + 1);
5 print "Name = $name<br>";
6 print "Domain = $domain";
7 ?>
```

ผลลัพธ์

Name = somsakkee
Domain = yahoo.com

นอกจากนี้ยังมีฟังก์ชัน strrpos() ที่ใช้สำหรับหาตำแหน่งข้อความหรือตัวอักษรเช่นเดียวกับฟังก์ชัน strpos() แต่จะเป็นการค้นหาโดยเริ่มจากตัวอักษรตัวสุดท้าย (จากขวาไปซ้าย) ผลลัพธ์ที่ได้จากฟังก์ชัน strrpos() จะเป็นตำแหน่งข้อความหรือตัวอักษรที่ต้องการค้นหา ที่เป็นลำดับสุดท้ายของข้อความหลัก ดูตัวอย่างการใช้ฟังก์ชัน strrpos() ด้านล่าง

ตัวอย่างที่ 6.14

การใช้ฟังก์ชัน strrpos()

```
1 <?php
2 $a = "123,456,789";
3 print strrpos($a, ",");
4 ?>
```

ผลลัพธ์

7

การหาความยาวของข้อความ (ฟังก์ชัน strlen())

ตัวอย่างที่ 6.15

การหาจำนวนตัวอักษรในข้อความด้วยฟังก์ชัน strlen()

```
1 <?php
2 $msg = "Puff, the magic dragon, lived by the sea.";
3 print "The length of &ldquo;$msg&rdquo; is ". strlen($msg) . " characters.";
4 ?>
```

ผลลัพธ์

The length of "Puff, the magic dragon, lived by the sea." is 41 characters.

การตัดช่องว่าง (ฟังก์ชัน ltrim(), rtrim(), trim(), chop())

- การตัดช่องว่างหน้าและท้ายข้อความทำได้โดยใช้ฟังก์ชัน trim() ดังตัวอย่างผลลัพธ์
- ฟังก์ชัน ltrim() ใช้ตัดช่องว่างเฉพาะข้างหน้าข้อความ
- ฟังก์ชัน rtrim() และ chop() ใช้ตัดช่องว่างเฉพาะด้านท้ายข้อความ

ตัวอย่างที่ 6.16

การใช้ฟังก์ชัน trim()

```
1 <?php
2 $msg = "      Puff, the magic dragon, lived by the sea.      ";
3 print "[Open]" . trim($msg) . "[Close]";
4 ?>
```

ผลลัพธ์

[Open]Puff, the magic dragon, lived by the sea.[Close]

การแบ่งแยกข้อความ (ฟังก์ชัน explode())

ฟังก์ชัน explode() ใช้สำหรับแบ่งข้อความออกเป็นคำหรือเป็นข้อความย่อย ๆ มีรูปแบบการใช้ดังนี้

รูปแบบ

array **explode** (string *separator*, string *string* [, int *limit*])

separator ตัวอักษรหรือข้อความที่ใช้เป็นตัวแบ่งแยกข้อความ
string ข้อความที่จะนำมาแยกออกเป็นข้อความย่อย
limit ตัวเลขระบุจำนวนข้อความย่อยสูงสุดที่ให้สร้างขึ้น จะมีหรือไม่ก็ได้

ตัวอย่างที่ 6.17

การใช้ฟังก์ชัน explode() เพื่อแยกชื่อและนามสกุลออกจากชื่อเต็ม

```
1 <?php
2 $fullname = "Albert Eistien";
3 list($firstname, $lastname) = explode(" ", $fullname);
4 print "First name : $firstname<br>";
5 print "Last name : $lastname";
6 ?>
```

ผลลัพธ์

First name : Albert
Last name : Eistien

จากตัวอย่าง ที่ 6.17 ในบรรทัดที่ 2 ฟังก์ชัน explode() จะแบ่งข้อความในตัวแปร \$fullname (โดยใช้ช่องว่างเป็นตัวแบ่ง) ออกได้เป็นสองข้อความ และเก็บข้อความทั้งสองไว้ในตัวแปร \$firstname และ \$lastname ตามลำดับ

ในกรณีที่มีการจำกัดจำนวนข้อความย่อยโดยการระบุค่า limit ฟังก์ชัน explode() จะทำการตัดข้อความจากซ้ายไปขวาจนได้จำนวนข้อความย่อยเท่ากับที่ระบุไว้ (limit) โดยข้อความส่วนที่เหลืออยู่ทางขวามือทั้งหมดจะเป็นข้อความย่อยอันสุดท้าย

ตัวอย่างที่ 6.18

การใช้ฟังก์ชัน `explode()` โดยกำหนดค่า `limit`

```
1 <?php
2 $fullname = "William Jefferson Clinton";
3 list($name1, $name2) = explode(" ", $fullname, 2);
4 print "Name1 : $name1<br>";
5 print "Name2 : $name2";
6 ?>
```

ผลลัพธ์

```
Name1 : William
Name2 : Jefferson Clinton
```

แต่หากไม่ได้ใส่ค่า `limit` ลงไป ข้อความในตัวแปร `$fullname` จะถูกแบ่งออกเป็น 3 ข้อความย่อย (เนื่องจากใช้ช่องว่างเป็นตัวแบ่งข้อความ และมีช่องว่างอยู่ 2 ตำแหน่งในข้อความนี้) โดยข้อความย่อยแรกจะถูกใส่ในตัวแปร `$name1` และข้อความย่อยที่สองจะถูกใส่ในตัวแปร `$name2` ส่วนข้อความย่อยที่สามจะไม่มีตัวแปรมารองรับค่าผลที่ได้จึงเป็นดังนี้

การรวมค่าในอาร์เรย์ให้เป็นสตริง (ฟังก์ชัน `implode()`)

ฟังก์ชัน `implode()` มีประโยชน์ในการนำค่าของแต่ละ `element` ในอาร์เรย์มาต่อกันเป็นข้อความเดียวกัน มีรูปแบบดังนี้

รูปแบบ

```
string implode (string glue, array pieces)
```

glue ตัวอักษร, ค่า หรือ ข้อความเชื่อม

pieces อาร์เรย์ของค่าหรือข้อความ

ตัวอย่างที่ 6.19

การรวมค่าในอาร์เรย์ให้เป็นสตริงด้วยฟังก์ชัน `implode()`

```
1 <?php
2 $ar_president = array('George', 'W.', 'Bush');
3 $fullname = implode(" ", $ar_president);
4 print $fullname;
5 ?>
```

ผลลัพธ์

```
George W. Bush
```

จากตัวอย่างเป็นการใช้ฟังก์ชัน `implode()` นำค่าของแต่ละ `element` ในตัวแปรอาร์เรย์ `$ar_president` มาเรียงต่อกัน โดยมีช่องว่างคั่นระหว่างแต่ละ `element` ผลลัพธ์ที่ได้จะถูกนำไปเก็บไว้ในตัวแปร `$fullname`

การแปลง case ของตัวอักษร (ฟังก์ชัน `strtolower()`, `strtoupper()`, `ucfirst()`, `ucword()`)

การแปลงข้อความเป็นตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กสามารถทำได้ 4 ลักษณะ โดยใช้ฟังก์ชันต่าง ๆ ดังนี้

- ฟังก์ชัน `strtolower()` ทำให้ข้อความเป็นตัวอักษรพิมพ์เล็กทั้งหมด
- ฟังก์ชัน `strtoupper()` ทำให้ข้อความเป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด
- ฟังก์ชัน `ucfirst()` ทำให้ตัวอักษรตัวแรกเป็นตัวอักษรพิมพ์ใหญ่
- ฟังก์ชัน `ucword()` ทำให้ตัวอักษรตัวแรกของคำแต่ละคำเป็นตัวอักษรพิมพ์ใหญ่

```

1 <?php
2 $imagine = "imagine all the people sharing all the world.";
3 print strtoupper($imagine . "<br>");
4 print ucfirst($imagine . "<br>");
5 print ucwords($imagine . "<br>");
6 $imagine2 = "Imagine All the People Sharing all the World.";
7 print strtolower($imagine2 . "<br>");
8 ?>

```

ผลลัพธ์

```

IMAGINE ALL THE PEOPLE SHARING ALL THE WORLD.
Imagine all the people sharing all the world.
Imagine All The People Sharing All The World.
imagine all the people sharing all the world.

```

การเปรียบเทียบสตริง (ฟังก์ชัน strcmp())

เมื่อต้องการเปรียบเทียบสตริงสองชุด เราสามารถใช้ฟังก์ชัน strcmp() ได้เช่นเดียวกับในภาษา C

รูปแบบ

```
int strcmp (string str1, string str2)
```

str1, str2

คือสตริงสองค่าที่นำมาเปรียบเทียบกัน

หาก str1 เท่ากับ str2 ฟังก์ชันจะคืนค่า 0 แต่หาก str1 น้อยกว่า str2 ฟังก์ชันจะคืนค่าลบ (-1) และหาก str1 มากกว่า str2 ฟังก์ชันจะคืนค่าบวก (1)

6.3 การทำงานกับตัวเลข

ดังที่กล่าวมาแล้วว่าในการสร้างตัวแปร เราไม่จำเป็นต้องกำหนดชนิด (type) ให้กับตัวแปร เนื่องจาก PHP จะกำหนดชนิดของตัวแปรให้โดยอัตโนมัติตามชนิดของข้อมูลที่ป้อนเข้าไป เช่น ในประโยค \$x = 5.13; ตัวแปร \$x จะถูกกำหนดเป็นชนิด double หรือในประโยค \$x = 5; ตัวแปร \$x จะถูกกำหนดเป็นชนิด integer นอกจากนี้ในประโยค \$x = "5.13"; ตัวแปร \$x จะถูกกำหนดให้เป็นชนิด string จะเห็นว่าความยืดหยุ่นในการจัดการข้อมูลของ PHP อาจสร้างความสับสนให้กับเราได้เช่นกัน ดังนั้น อันดับแรกที่เราจะต้องทราบเกี่ยวกับการจัดการข้อมูลที่เป็นตัวเลขคือ การตรวจสอบชนิดของข้อมูลหรือตัวแปร

การตรวจสอบชนิดของข้อมูลหรือตัวแปร

ฟังก์ชันใน PHP ที่ใช้สำหรับตรวจสอบตัวแปรที่เกี่ยวกับตัวเลข ได้แก่ ฟังก์ชัน is_int() หรือ is_integer() ใช้ตรวจสอบว่าตัวแปรเป็นชนิด integer หรือไม่ และฟังก์ชัน is_float() หรือ is_double() ใช้ตรวจสอบว่าตัวแปรเป็นชนิด double หรือ float หรือไม่ นอกจากนี้ยังมีฟังก์ชัน gettype() ที่ให้ผลลัพธ์เป็นชื่อชนิดของตัวแปรนั้น ๆ ดังตัวอย่าง

ตัวอย่างที่ 6.21	การตรวจสอบชนิดของตัวแปรด้วยฟังก์ชัน <code>gettype()</code>
<pre> 1 <?php 2 \$a = 455; 3 \$b = 3.1745; 4 \$c = "10.10"; 5 \$z = \$a + \$b + \$c; 6 print "\\$a = \$a is " . gettype(\$a) . "
"; 7 print "\\$b = \$b is " . gettype(\$b) . "
"; 8 print "\\$c = \$c is " . gettype(\$c) . "
"; 9 print "\\$z = \$z is " . gettype(\$z) . "
"; 10 11 if (is_int(\$z) is_float(\$z)) { 12 print "The result is a number."; 13 } else { 14 print "The result is not a number."; 15 } 16 ?> </pre>	ผลลัพธ์ <pre> \$a = 455 is integer \$b = 3.1745 is double \$c = 10.10 is string \$z = 468.2745 is double The result is a number. </pre>

การปัดเศษทศนิยม

PHP มีฟังก์ชัน 3 ฟังก์ชันที่ใช้สำหรับปัดเศษทศนิยม ด้วยเงื่อนไขที่แตกต่างกันตามแต่ความต้องการในการใช้งาน นั่นคือ `floor()`, `ceil()` และ `round()`

- การปัดเศษทิ้ง เมื่อต้องการปัดเศษทิ้งทั้งหมด ให้คงเหลือไว้เฉพาะเลขหน้าจุดทศนิยมเท่านั้น จะใช้ฟังก์ชัน `floor()` ดังตัวอย่าง

ตัวอย่างที่ 6.22	การปัดเศษทศนิยมทิ้งด้วยฟังก์ชัน <code>floor()</code>
<pre> 1 <?php 2 \$num1 = 321.1234; 3 \$num2 = 321.9876; 4 print "floor(\$num1) = " . floor(\$num1) . "
"; 5 print "floor(\$num2) = " . floor(\$num2) . "
"; 6 ?> </pre>	ผลลัพธ์ <pre> floor(321.1234) = 321 floor(321.9876) = 321 </pre>

- การปัดเศษขึ้น เมื่อต้องการปัดเลขที่มีทศนิยม (ไม่ว่าทศนิยมจะสูงหรือต่ำกว่า 0.5) ขึ้นเป็นเลขจำนวนเต็ม จะใช้ฟังก์ชัน `ceil()`

ตัวอย่างที่ 6.23	การปัดเศษทศนิยมขึ้นด้วยฟังก์ชัน <code>ceil()</code>
<pre> 1 <?php 2 \$num1 = 321.1234; 3 \$num2 = 321.9876; 4 print "ceil(\$num1) = " . ceil(\$num1) . "
"; 5 print "ceil(\$num2) = " . ceil(\$num2) . "
"; 6 ?> </pre>	ผลลัพธ์ <pre> ceil(321.1234) = 322 ceil(321.9876) = 322 </pre>

- การปัดเศษโดยระบุจำนวนตำแหน่งทศนิยม

ในกรณีที่ต้องการแปลงเลขทศนิยมเป็นเลขจำนวนเต็ม หรือเป็นเลขทศนิยมที่มีจำนวนตัวเลขหลังจุดทศนิยมตามที่ต้องการ เรา จะใช้ฟังก์ชัน `round()` ซึ่งมีรูปแบบดังนี้

รูปแบบ

```
float round (float val [, int precision])
```

val คือตัวเลขที่มีทศนิยมที่ต้องการนำมาปัดเศษ

precision คือจำนวนตัวเลขหลังจุดทศนิยมที่ต้องการ สามารถระบุเป็นค่า 0, ค่าบวก หรือ ค่าลบก็ได้ ส่วนค่า default เมื่อไม่มีการระบุจะหมายถึงค่า 0 ค่าที่เป็นบวกจะทำให้เลขที่อยู่หลังจุดทศนิยมถูกปัดเศษ ส่วนค่าเป็นลบจะทำให้เลขที่อยู่หน้าจุดทศนิยมถูกปัดเศษ

ตัวอย่างที่ 6.24	การปัดเศษทศนิยมโดยกำหนดให้มีตัวเลขหลังจุดทศนิยม 4 หลักด้วยฟังก์ชัน round()	
<pre> 1 <?php 2 \$price1 = 1030; 3 \$price2 = 1027; 4 \$price3 = 999; 5 \$AveragePrice = (\$price1 + \$price2 + \$price3) / 3; 6 print "Average Price = \$AveragePrice
"; 7 print "Round Average Price = " . round(\$AveragePrice, 4); 8 ?> </pre>	ผลลัพธ์	<pre> Average Price = 1018.6666666667 Round Average Price = 1018.6667 </pre>

ตัวอย่างที่ 6.25	การปัดเศษทศนิยมโดยกำหนดให้มีตัวเลขหลังจุดทศนิยม 4 หลักด้วยฟังก์ชัน round()	
<pre> 1 <?php 2 print round(65456587, -6); 3 ?> </pre>	ผลลัพธ์	<pre> 65000000 </pre>

6.4 การทำงานกับวันที่และเวลา

การทำงานกับวันที่และเวลาใน PHP มักจะเกี่ยวข้องกับรูปแบบต่าง ๆ ของวันที่และเวลา และการดึงค่าวันที่ ณ ปัจจุบันของเครื่องเซิร์ฟเวอร์ รวมถึงการตรวจสอบความถูกต้องของวันที่ด้วยฟังก์ชัน checkdate()

วันที่และเวลาปัจจุบัน

ในการเขียนโปรแกรมโดยทั่วไปที่เกี่ยวข้องกับการบันทึกหรือแสดงข้อมูลที่เป็นวันที่ มักพบเสมอว่ามีความต้องการใช้ฟังก์ชันที่สามารถบอกถึงวันที่และเวลา ณ ปัจจุบันของเครื่องคอมพิวเตอร์ได้ ซึ่งใน PHP มีอยู่หลายฟังก์ชันที่ให้ผลลัพธ์เช่นนี้ได้ เช่น ฟังก์ชัน date() และ getdate() แต่สิ่งที่แตกต่างกันระหว่างฟังก์ชันทั้งสองคือผลลัพธ์ที่ได้จากฟังก์ชัน date() เป็นสตริง และเราสามารถกำหนดรูปแบบของวันที่และเวลาได้โดยตรง ส่วนฟังก์ชัน getdate() จะให้ผลลัพธ์ที่เป็นอาร์เรย์ของ วัน เดือน ปี และเวลาในรูปแบบต่าง ๆ

รูปแบบการใช้ฟังก์ชัน date()

รูปแบบ

string date (string format [, int timestamp])

format คือรูปแบบของวันที่และเวลาซึ่งจะแทนด้วยตัวอักษรภาษาอังกฤษหนึ่งตัว

timestamp เป็นตัวเลข integer ลำดับเวลาทุกหนึ่งวินาที นับตั้งแต่วันที่ 1 เดือนมกราคม ปี ค.ศ.1970 เวลา 00:00:00 น.

ตารางแสดงตัวอักษรที่ใช้กำหนดรูปแบบวันที่ในพารามิเตอร์ format

ตัวอักษร	ความหมาย	ตัวอย่างผลที่ได้รับ
a	แสดงเป็น Ante meridiem และ Post meridiem ตัวพิมพ์เล็ก	am หรือ pm
A	แสดงเป็น Ante meridiem และ Post meridiem ตัวพิมพ์ใหญ่	AM หรือ PM
c	วันที่แบบ ISO 8601	2004-02-12T15:19:21+00:00
d	วันที่ในหนึ่งเดือน ประกอบด้วยตัวเลข 2 หลัก (หากเป็นวันที่น้อยกว่า 10 จะนำหน้าด้วยเลข 0	01 ถึง 31
D	วันในหนึ่งสัปดาห์ โดยแสดงเป็นตัวอักษรภาษาอังกฤษ 3 ตัว	Mon ถึง Sun
F	ชื่อเดือนในภาษาอังกฤษ	January ถึง December
g	เวลาในรูปแบบ 12 ชั่วโมง โดยไม่มีเลข 0 นำหน้า	1 ถึง 12
G	เวลาในรูปแบบ 24 ชั่วโมง โดยไม่มีเลข 0 นำหน้า	0 ถึง 23
h	เวลาในรูปแบบ 12 ชั่วโมง โดยมีเลข 0 นำหน้า	01 ถึง 12
H	เวลาในรูปแบบ 24 ชั่วโมง โดยมีเลข 0 นำหน้า	00 ถึง 23
i	เลขนาทีย่อยโดยมีเลข 0 นำหน้า	00 ถึง 59
j	วันที่โดยไม่มีเลข 0 นำหน้า	1 ถึง 31
l (แอลตัวเล็ก)	ชื่อวัน (แบบเต็ม) ในภาษาอังกฤษ	Sunday ถึง Saturday
L	แสดงค่าที่บอกว่าเป็นปีอธิกสุรทิน (ปีที่เดือนกุมภาพันธ์มี 29 วัน) หรือไม่	ถ้าเป็นปีอธิกสุรทินจะให้ค่า 1 หากไม่ใช่ให้ค่าเป็น 0
m	แสดงเลขเดือนโดยมีเลข 0 นำหน้า	01 ถึง 12
M	แสดงชื่อเดือนในภาษาอังกฤษเป็นตัวย่อ 3 ตัวอักษร	Jan ถึง Dec
n	แสดงเลขเดือนโดยไม่มีเลข 0 นำหน้า	1 ถึง 12
r	วันที่ในรูปแบบ RFC 2822	Thu, 21 Dec 2000 16:01:07+0200
s	แสดงเลขวินาทีโดยมีเลข 0 นำหน้า	00 ถึง 59
S	แสดงส่วนต่อท้าย (suffix) วันที่ในภาษาอังกฤษ	st, nd, rd หรือ th
w	แสดงตัวเลขของวันในสัปดาห์	0(วันอาทิตย์) ถึง 6 (วันเสาร์)
Y	แสดงปี ค.ศ. เป็นเลขสี่หลัก	1998, 1999, 2004, 2005
y	แสดงปี ค.ศ. เป็นเลขสองหลัก	98, 99, 04, 05

ฟังก์ชัน `getdate()` มีรูปแบบการใช้งานดังนี้

รูปแบบ

`array getdate ( [int timestamp] )`

timestamp เป็นตัวเลข integer ลำดับเวลาทุกหนึ่งวินาที นับตั้งแต่วันที่ 1 มกราคม ปีค.ศ. 1970

ตัวอย่างที่ 6.26

การใช้ฟังก์ชัน `date()` และ `getdate()`

```

1 <?php
2 print date("d-m-Y") . "<br>";
3 print date("d-m-Y H:i:s") . "<br>";
4 print date("d F Y h:i:s a") . "<br>";
5 print date("l F d Y H:i") . "<br><br>";
6
7 $mNow = getdate();
8 $xNow = $mNow['mday'] . "-" . $mNow['mon'] . "-" .
9         $mNow['year'] . " " . $mNow['hour'] . ":" .
10        $mNow['minutes'] . ":" . $mNow['second'];
11 print $xNow . "<br>";
12 ?>

```

ผลลัพธ์

```

05-08-2007
05-08-2007 14:26:50
05 August 2007 02:26:50 pm
Sunday August 05 2007 14:26

5-8-2007 :26:

```

การตรวจสอบความถูกต้องของวันที่

ในการบันทึกข้อมูลไม่ว่าจะเป็นแบบสอบถาม การสมัครสมาชิก การลงทะเบียน มักมีข้อมูลวันที่เข้าไปเกี่ยวข้องด้วยเสมอ แต่ในระบบฐานข้อมูลเช่น MySQL จะมีความยืดหยุ่นสูงในการบันทึกข้อมูลประเภทวันที่ จึงเป็นหน้าที่ของผู้เขียนโปรแกรมที่จะต้องตรวจสอบความถูกต้องของข้อมูลวันที่ ก่อนทำการบันทึกลงฐานข้อมูล (รูปแบบ วันที่ที่ MySQL รับได้คือ yyyy-mm-dd, yy-mm-dd)

ใน PHP มีฟังก์ชัน `checkdate()` ที่ทำหน้าที่ตรวจสอบความถูกต้องของข้อมูลประเภทวันที่ โดยมีเงื่อนไขการตรวจสอบคือ

- ปี ค.ศ. ตั้งแต่ 1 ถึง 32767
- เดือนตั้งแต่ 1 ถึง 12
- วันที่ ที่ถูกต้องของเดือนนั้น ๆ ซึ่งจะมีการตรวจสอบในกรณีที่ปีเป็นปีอธิกสุรทิน

รูปแบบ

```
bool checkdate (int month, int day, int year)
```

month, day, year

เดือน, วัน, ปี ตามลำดับ

ตัวอย่างที่ 6.27

การตรวจสอบความถูกต้องของวันที่ด้วยฟังก์ชัน `checkdate()`

```

1 <?php
2 $mdate = "29/2/2002";
3 list($mday, $mmonth, $myear) = explode("/", $mdate);
4 if (checkdate($mmonth, $mday, $myear)) {
5     print "The date $mdate is valid.";
6 } else {
7     print "$mdate is invalid date!";
8 }
9 ?>

```

ผลลัพธ์

```
29/2/2002 is invalid date!
```

6.5 การใช้ฟอร์มติดต่อไปยังไฟล์ PHP

First name : → **fname**
Last name : → **lname**
Age : → **age**
Sex : → **sex**

จากตัวอย่าง เป็นฟอร์มชื่อ Regis.php ซึ่ง ขอบเขตของฟอร์มชื่อ form1 และมีช่องกรอกข้อมูลชื่อ fname, lname, age, sex ตามลำดับ ใน code ของ Regis.php ส่วนของแท็ก **<form>** ให้ทำการใส่คุณสมบัติ **action** โดยให้ระบุไฟล์ php ที่ จะทำการส่งข้อมูลไปจาก ฟอร์ม ซึ่งในตัวอย่างนี้ให้ทำการส่งข้อมูลไปยังไฟล์ php ชื่อ “registverify.php” ซึ่งสามารถระบุได้ดังนี้

<form name="form1" method="post" action="registverify.php">

โปรแกรมด้านล่างนี้เป็นไฟล์ชื่อ registverify.php ซึ่งเป็นไฟล์ php ที่ทำการรับข้อมูลมาจากฟอร์มที่ชื่อ Regis.php โดยภายใน ไฟล์นี้จะมองช่องกรอกข้อมูลจากไฟล์ Regis.php เป็นตัวแปร โดยจะมองเป็นตัวแปร \$fname, \$lname, \$age, \$sex ตามลำดับ ซึ่งเราสามารถนำตัวแปรเหล่านี้ ไป ประมวลผล หรือ insert ลงฐานข้อมูลได้

```
<?php
print "<center><h3>";
if ($age < 20) {
    print "น้อง $fname สวัสดีครับ";
} elseif ($age <= 30) {
    print "สวัสดีครับ คุณ$fname $lname";
} elseif ($age <= 40) {
    print "คุณพี่ $fname $lname สวัสดีครับ";
} else {
    if ($sex == "Female") {
        print "สวัสดีครับ คุณแม่$fname $lname";
    } else {
        print "สวัสดีครับ คุณลุง$fname $lname";
    }
}
print "</h3></center>";
?>
```

ผลการรัน หลังจากการกดปุ่ม submit จากไฟล์ Regis.php

สวัสดีครับ คุณVinona Rider