

Chapter 4

PHP Operation

954242 Web Programming
Sec.002 Semester 02/2555

Sakol Liamprawat
sakolcnx@gmail.com

วัตถุประสงค์

- สามารถอธิบายหลักการทำงานและการใช้คำสั่งเบื้องต้นของภาษา PHP
- สามารถอธิบายการเขียนค่าสตริงรูปแบบต่างๆ และการทำงานกับตัวอักษรสตริง และการใช้งานฟังก์ชันต่างๆเพื่อจัดการกับสตริงในภาษา PHP
- สามารถอธิบายการทำงานกับเลขจำนวน และการใช้ฟังก์ชันต่างๆกับเลขจำนวนในภาษา PHP
- สามารถอธิบายวิธีการจัดเก็บ การแสดงผล และการดำเนินการกับข้อมูล/วันเวลาในภาษา PHP

บทนำ

- การใช้ภาษา HTML เป็นการนำเสนอผ่านโปรแกรม Web Browser มีขีดจำกัด เพราะข้อมูลใน Web Page เป็นข้อมูลที่คงที่ หรือ Static (ถูกสร้างหรือแก้ไขจนเสร็จแล้วจึงนำเสนอ)
- ดังนั้นเมื่อต้องการปรับปรุงแก้ไข (Update) ข้อมูลไฟล์ HTML ผู้สร้าง Web Page ต้องเข้าไปแก้ไขไฟล์นั้นๆ โดยตรง เป็นอุปสรรคทำให้การทำงานเป็นไปด้วยความล่าช้า
- ภาษา PHP เป็นหนึ่งในภาษาที่ถูกพัฒนาขึ้นมาแก้ไขปัญหาที่กล่าวมา และเป็นภาษาที่ได้รับความนิยมสูงมากสำหรับนักพัฒนาโปรแกรมบนเว็บ
- Web Page คือไฟล์เอกสารในภาษา HTML มีการแสดงผลในรูปแบบที่ต้องการ
- Script หรือไฟล์สคริปต์ เป็นไฟล์ที่มีการแทรกคำสั่งด้วยภาษา php เข้าไปยัง Web Page ปกติ
- Web Application คือการนำ Web Page และไฟล์ Script หลากๆ ไฟล์มาจัดเรียงเรียงให้ทำงานตามวัตถุประสงค์ที่ต้องการ เช่น ระบบลงทะเบียน ระบบซื้อขายสินค้าออนไลน์ ฯลฯ

รูปแบบของภาษา PHP

- ภาษา PHP เป็นภาษาสคริปต์ภาษาหนึ่งที่ทำเนิการที่ฝั่งเซิร์ฟเวอร์ (Server-side) หรือ Server-side script คือเมื่อ Code ถูกเรียกใช้โดย Web Browser โปรแกรม PHP ที่อยู่ในเครื่องที่เป็น Web Server จะทำการประมวลผลแล้วสร้าง (Generate) ผลลัพธ์ที่อยู่ในรูปของภาษา HTML ขึ้นแล้วจึงส่งมาให้กับ Client เพื่อให้ Web Browser แสดงผล

Server side

```
<html>
<head>
<title> PHP Script
</title>
</head>
<body>
<?php
  echo "Sawasdee";
?>
</body>
</html>
```

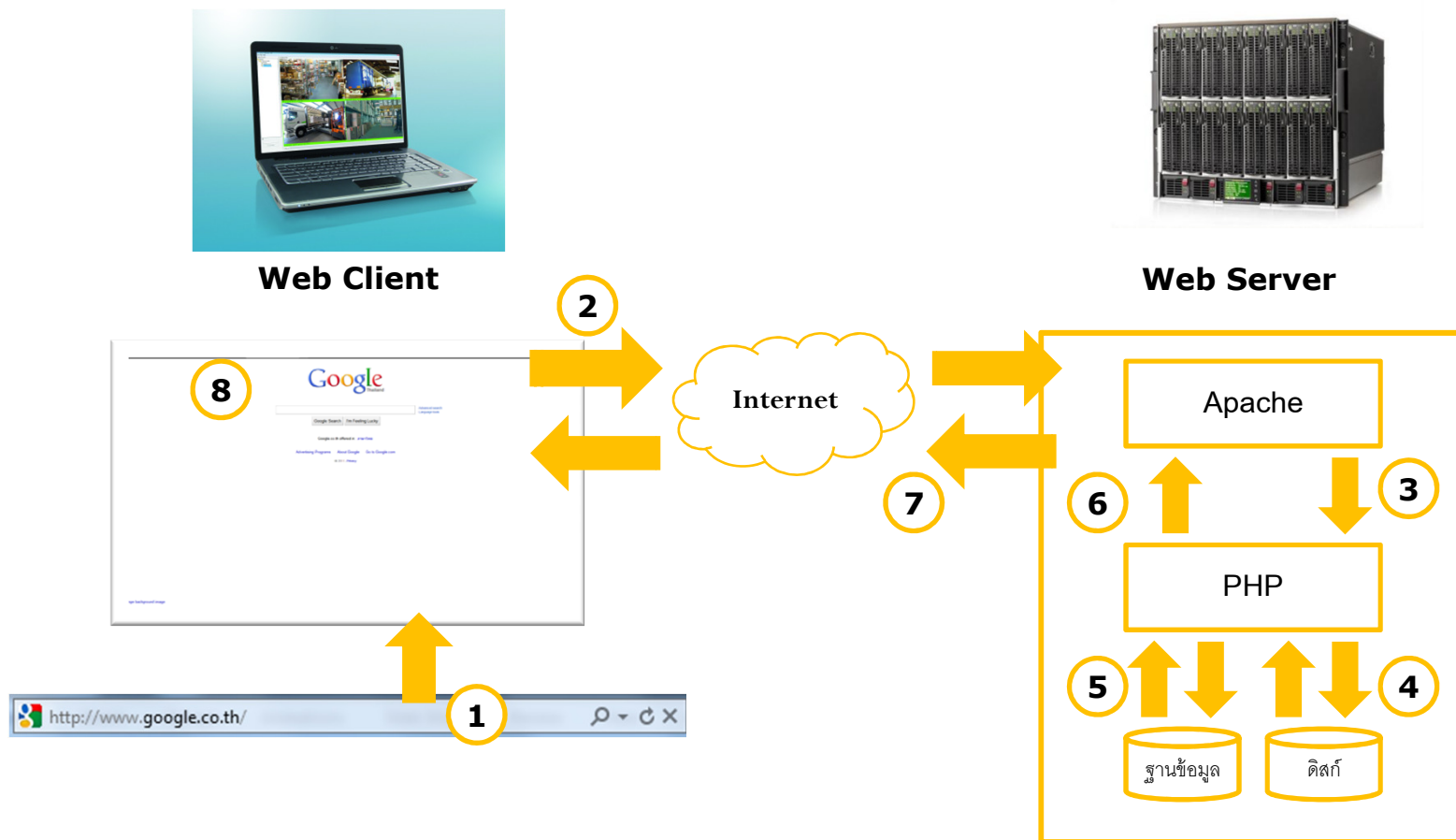
Client side

```
<html>
<head>
<title> PHP Script
</title>
</head>
<body>
  Sawasdee
</body>
</html>
```

Processed

การทำงานของภาษา PHP

การทำงานทั้งหมดที่เกิดขึ้นเมื่อ Web Server เรียกไปยัง Web Page ที่มีคำสั่งภาษา PHP ฝังอยู่สามารถอธิบายได้ดังรูป



การทำงานของภาษา PHP

1. ผู้ใช้พิมพ์ชื่อ Web Page ที่มีคำสั่งภาษา PHP ฝังอยู่ ลงในช่อง Address หรือช่อง URL ของโปรแกรม Web Browser
2. Web Browser ส่งข้อความผ่านอินเทอร์เน็ตไปยังเครื่องคอมพิวเตอร์ Web Server เพื่อร้องขอหน้า Web Page นั้นๆ (ไฟล์ php)
3. โปรแกรม Web Server ที่ทำงานบนเครื่อง Server (สมมติว่าเป็นโปรแกรม Apache) เมื่อได้รับข้อความที่ส่งมาก็จะขอให้ตัวแปลภาษา PHP (PHP Interpreter) ซึ่งเป็นโปรแกรมที่ทำงานบนเครื่อง Server เช่นเดียวกัน ทำการประมวลผลคำสั่ง PHP ที่มีอยู่ในหน้าเพจนั้นๆ
4. PHP Interpreter อ่านเนื้อหาของไฟล์ PHP ขึ้นมาจากดิสก์
5. PHP Interpreter ทำงานไปตามคำสั่ง PHP ในไฟล์ ซึ่งอาจมีการติดต่อไปยังฐานข้อมูลเพื่อดึงข้อมูลมาใช้งานหรือส่งข้อมูลไปเก็บด้วย (ขึ้นอยู่กับว่าคำสั่ง PHP ในไฟล์กำหนดว่าให้ทำอะไรบ้าง)
6. PHP Interpreter ส่งผลลัพธ์ที่ได้จากไฟล์ PHP ไปให้ Web Server ในขั้นตอนที่ 3 ผลลัพธ์นี้จะเป็นคำสั่งภาษา HTML (HTML Tags) โดยไม่มีคำสั่งในภาษา PHP รวมอยู่ด้วยแต่อย่างใด
7. โปรแกรม Web Server ส่งเนื้อหาที่ได้รับมาจาก PHP Interpreter กลับไปให้โปรแกรม Web Browser ในเครื่องของผู้ใช้ เพื่อเป็นการตอบสนอง ต่อคำร้องขอของเว็บเบราว์เซอร์ (ฝั่ง Web Client)
8. Web Browser ตีความเนื้อหาของนั้นแล้วแสดงผลออกมาบนหน้าจอ ตามคำสั่ง HTML ที่กำหนดไว้ในเนื้อหาของนั้น

PHP – Dynamic Language

- PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ
- PHP เป็นเครื่องมือที่ช่วยให้สามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและหลากหลายมากขึ้น
- PHP คือ ภาษาโปรแกรมซึ่งไม่เหมือนกับ HTML ที่เป็นเพียงภาษาสำหรับอธิบายหน้าเอกสารโดย PHP Interpreter ที่ทำงานอยู่ในเครื่องเว็บเซิร์ฟเวอร์จะอ่านคำสั่งที่เขียนด้วยภาษา PHP แล้วทำงานตามคำสั่งเหล่านั้น เช่นเป็นการเก็บค่าไว้ในตัวแปร การตัดสินใจเลือกทำงานโดยขึ้นอยู่กับเงื่อนไขบางอย่าง การทำซ้ำ (วนลูป) หรือ อาจเป็นการทำงานที่ซับซ้อนขึ้น เช่น การอ่านไฟล์ข้อมูล เขียนข้อมูลลงไฟล์ ติดต่อกับฐานข้อมูล หรือรับ-ส่งอีเมล เป็นต้น

สคริปต์เบื้องต้นของภาษา PHP

การเขียนภาษา PHP นั้น

- จะเริ่มต้นด้วยแท็ก `<?php`
- และปิดท้ายด้วย `?>`

```
<?php  
    echo "Sawasdee";  
?>
```

- การเขียน PHP สามารถละแท็ก `<html>`, `<head>` และ `<body>` ได้
- ภายในบริเวณ Tag PHP ต้องเขียนภาษา PHP เท่านั้น ไม่สามารถเขียน HTML ลงไปได้ เพราะ PHP จะแปลไม่ออก

```
<?php  
    <p>Sawasdee</p>  
?>
```



การแสดงความสวยงามบนจอภาพ

- การแสดงความงามไปที่ Web Browser ใช้ฟังก์ชัน print หรือ echo เช่น

```
<?php
```

```
    echo "Hello";
```

```
?>
```

หรือเขียนไว้บรรทัดเดียวกัน

```
<?php echo "Hello"; ?>
```

หรือลดรูปโดยตัดคำว่า php และ echo ออก

<?= "Hello" ?> แต่มีข้อเสียคือไม่สามารถเขียนคำสั่ง PHP ได้ ใช้งานได้

เฉพาะแสดงความงามหรือนิพจน์เท่านั้น เช่น <?= 1+2 ?>

- echo ต่างจาก print ที่สามารถแทรกชื่อตัวแปรเข้าไปอยู่ในข้อความได้ทันที (ข้อความจะต้องครอบด้วยเครื่องหมาย “ ”)

```
echo "x+y=" . ($x+$y) . "<br />";  
echo "(x-y)/2=" . ($x-$y)/2 . "<br />";
```

รูปแบบ

echo ข้อความ;

หรือ

echo \$ชื่อตัวแปร

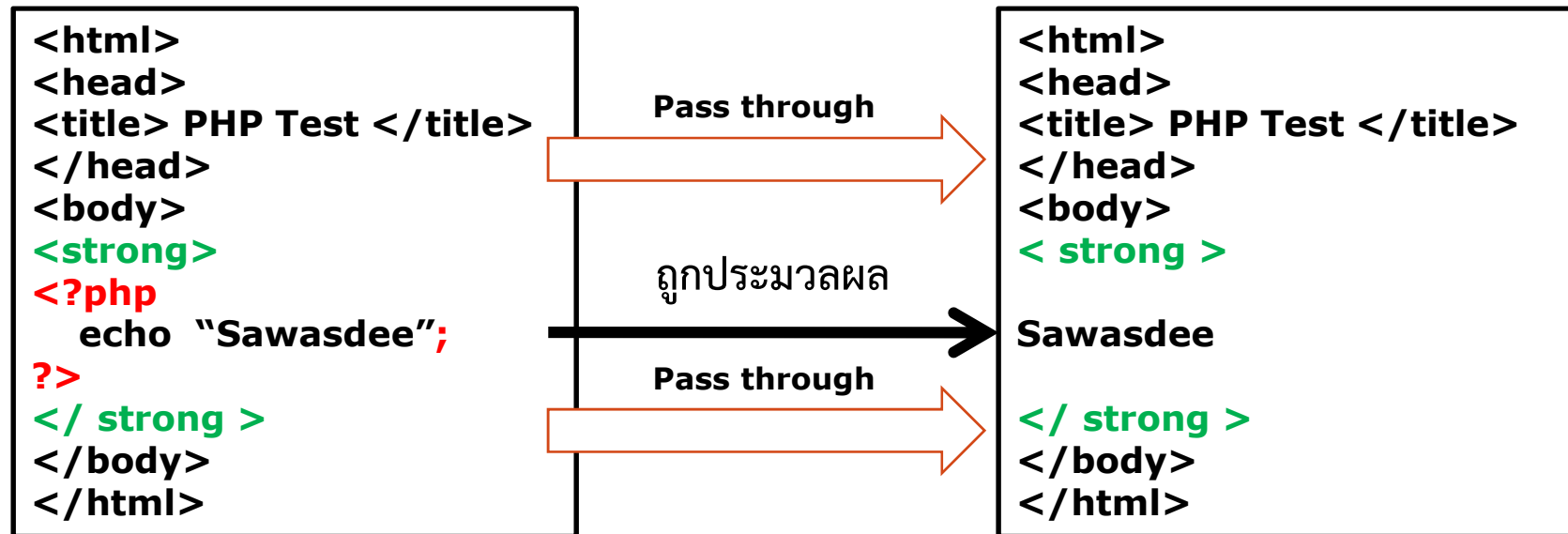
ภาษา PHP เบื้องต้น

- ภาษา PHP มีลักษณะคล้ายกับภาษาซี ที่คำสั่งแต่ละคำสั่งนั้นจะต้องปิดท้ายด้วยเครื่องหมาย Semi colon (;) ทำหน้าที่เพื่อแยกคำสั่งแต่ละคำสั่งออกจากกัน

```
<html>
  <head>
    <title> PHP Script </title>
  </head>
  <body>
    <?php
      echo "Sawasdee";
      echo "<br />";
      echo "Good bye";
    ?>
  </body>
</html>
```

ภาษา PHP เบื้องต้น

- ภาษา PHP สามารถสอดแทรกไว้ในภาษา HTML ได้ใน **ทุกส่วนหรือทุกตำแหน่ง** ที่ต้องการ



- คำสั่งทั้งหลายที่อยู่นอกแท็ก PHP จะถูก **ส่งผ่าน (Pass through)** โดยไม่มีการเปลี่ยนแปลงแก้ไขหรือแม้แต่จัดรูปแบบใดๆทั้งสิ้น

ตำแหน่งของสคริปต์ PHP

- Tag ของ PHP ไม่จำเป็นต้องอยู่ภายในส่วน <body> ของ Web Page
- จะมี Tag PHP จำนวนเท่าไรก็ได้ภายใน Web Page

```
<html>
<head>
<title><?php echo "PHP Test"; ?></title>
</head>
<body>
<?php
echo "<table border='1' width='100%'>";
echo "<tr><td>Sawassdee</td><td>PHP</td></tr>";
echo "</table>";
?>
</body>
</html>
```

PHP – Dynamic Language

- ความสำคัญของภาษา PHP อยู่ที่การสร้างเนื้อหาส่วนใดส่วนหนึ่งหรือทั้งหมดของเว็บเพจขึ้นมาเป็นแบบพลวัต หรือ ไดนามิก (Dynamic)

```
<html>
<head>
<title> Date and time dynamic</title>
</head>
<body>
<?php
    echo "It is ";
    echo date("H : i : s");
    echo " of date ";
    echo date("j F Y");
?>
</body>
</html>
```

ประมวลผล

รอบ 1

It is 18 : 58 : 51 of date 2 June 2011

It is 19 : 00 : 21 of date 2 June 2011

ประมวลผลรอบ 2

(เมื่อเวลาผ่านไป)

PHP – Whitespace

- PHP Interpreter ไม่สนใจการขึ้นบรรทัดใหม่, ช่องว่าง และแท็บ ซึ่งรวมเรียกว่า **Whitespace** โดย PHP Interpreter จะดูจากตำแหน่งของเครื่องหมาย semicolon เป็นหลัก เพื่อตัดสินว่าประโยคคำสั่งต่างๆ เริ่มต้นตรงไหน และสิ้นสุดตรงไหน

```
<html>
<head>
<title> Date and time dynamic</title>
</head>
<body>
<?php
    echo "It is ";          echo date( "H:i:s" );
    echo " of date ";
    echo date(
        "j F Y"
    );
?>
</body>
</html>
```

คอมเมนต์ในภาษา PHP

- ในการเขียน Code PHP สามารถเขียนคำอธิบายสคริปต์ หรือคอมเมนต์ (Comment) กำกับไว้เพื่ออธิบายว่าแต่ละส่วนใช้ทำอะไร ซึ่งจะเป็นประโยชน์สำหรับผู้อื่นที่จะนำไปพัฒนาต่อ (HTML ใช้ `<!-- - กับ - ->`)
- PHP Interpreter จะไม่สนใจคอมเมนต์ คือ ไม่นำส่วนของคอมเมนต์มาตีความและประมวลผล ถึงแม้ว่าสิ่งที่อยู่ในคอมเมนต์จะเป็น Code PHP ดังนั้นประโยชน์คือ ใช้ปิดการทำงานของ Code ส่วนใดส่วนหนึ่งใน Web Page ไว้ชั่วคราว แทนที่จะทำการลบ Code ส่วนนั้นทิ้งไปเลย

รูปแบบคอมเมนต์	สัญลักษณ์ที่ใช้	คำอธิบาย
คอมเมนต์แบบ <u>บรรทัดเดียว</u>	// หรือ #	ข้อความใดๆตั้งแต่เครื่องหมายนี้ไปจนสุดบรรทัด จะกลายเป็นคอมเมนต์
คอมเมนต์แบบ <u>หลายบรรทัด</u>	/* */	ข้อความที่อยู่ระหว่าง /* กับ */ จะกลายเป็นคอมเมนต์

คอมเมนต์ในภาษา PHP

```
<html>
<head>
<title> Date and time dynamic</title>
</head>
<body>
<?php
/* โปรแกรมนี้จะแสดงวันและเวลาปัจจุบันของเครื่อง Server
ออกมาในหน้าจอของโปรแกรม Web Browser */
    echo "It is ";
    echo date("H : i : s");      // แสดงเวลาปัจจุบันออกมาในรูปแบบ "ชั่วโมง : นาที : วินาที"
    echo " of date ";
    echo date("j F Y");        // แสดงวันปัจจุบันออกมาในรูปแบบ "วันที่ เดือน ปี"
?>
</body>
</html>
```

การประกาศตัวแปร

- ข้อมูลเป็นสิ่งสำคัญของ Programming เพราะต้องเขียนสคริปต์ PHP เพื่อนำข้อมูลไปประมวลผล
- ระหว่างประมวลผล ต้องใช้ตัวแปร (Variable) ทำหน้าที่เก็บค่าของข้อมูลนั้น
- การตั้งชื่อตัวแปร เพื่อแยกแยะ บ่งชี้ และเข้าใจการทำงาน
- ก่อนจะใช้งานตัวแปร ต้องประกาศตัวแปร (Variable Declaration) ให้ PHP Engine ทราบก่อน

\$ชื่อตัวแปร = ค่าตัวแปร; เช่น \$a=5;

- จะต้องขึ้นต้นด้วยเครื่องหมาย \$ (Dollar sign)
- ตามด้วยชื่อตัวแปร และค่าตัวแปร
- ข้อมูลและตัวแปรชนิด Null หมายถึงการระบุความไม่มีค่าใดๆ อยู่ในตัวแปร (ไม่ใช่เท่ากับศูนย์) คือ ว่างเปล่า

\$ชื่อตัวแปร = Null; เช่น \$a=Null;

การตั้งชื่อตัวแปร ในภาษา PHP

- มีสัญลักษณ์ \$ นำหน้าสุดเสมอ
- ชื่อตัวแปรในภาษา PHP จะต้องประกอบไปด้วย ตัวอักษร (อังกฤษ หรือ ไทย) ตัวเลข หรือ เครื่องหมายขีดกลาง (Underscore) เท่านั้น
- ชื่อตัวแปรห้ามขึ้นต้นด้วย ตัวเลข และห้ามมีสัญลักษณ์ทางคณิตศาสตร์ประกอบ
- ใช้สัญลักษณ์ _ (underscore) ขึ้นต้นหรือผสมในชื่อตัวแปรได้ แต่ห้ามมีเว้นวรรค

ชื่อตัวแปรที่ใช้ได้
\$N_park
\$Funny555
\$S_and_P

ชื่อตัวแปรที่ใช้ไม่ได้	สาเหตุที่ใช้ไม่ได้
\$N-park	ชื่อตัวแปรมีเครื่องหมายขีดตรงกลาง
\$555Funny	ชื่อตัวแปรขึ้นต้นด้วยตัวเลข
\$S&P	ชื่อตัวแปรมีเครื่องหมาย &

!!! ภาษา PHP ถือว่าตัวอักษรเล็กกับตัวอักษรใหญ่ในชื่อตัวแปรแตกต่างกัน (Case Sensitive) !!!

ชนิดของข้อมูล (Data Type)

- ตัวแปรแต่ละตัวจะมีชนิดข้อมูลประจำตัว โดยจะขึ้นอยู่กับชนิดข้อมูลที่ถูกเก็บไว้

ชนิดข้อมูล	คำอธิบาย
<u>Scalar type</u>	
Boolean (ตรรกศาสตร์)	เป็นชนิดข้อมูลที่เป็น ค่าจริง (True) หรือ ค่าเท็จ (False) ใดๆอย่างหนึ่ง
Integer (เลขจำนวนเต็ม)	เป็นชนิดข้อมูลค่าเลขจำนวนเต็ม (เต็มบวก, เต็มลบ หรือ ศูนย์)
Float or Double (เลขจำนวนทศนิยม)	เป็นชนิดข้อมูลของจำนวนจริง
String (สตริง)	เป็นชนิดของข้อมูลกลุ่มอักษรที่ไม่มีขอบเขตจำกัด
<u>Compound type</u>	
Array (อาร์เรย์)	เป็นชนิดของกลุ่มข้อมูลที่เกี่ยวข้งกัน ถูกจัดเก็บไว้ในตัวแปรหนึ่งๆ
Object (ออบเจ็ค)	เป็นชนิดของกลุ่มที่รวมไปด้วยข้อมูลและโค้ดที่ใช้ทำงานกับข้อมูลนั้นๆ
<u>Special type</u>	
Resource	เป็นชนิดข้อมูลพิเศษที่ใช้อ้างอิงถึงทรัพยากรภายนอก โดยใช้ฟังก์ชันพิเศษเป็นตัวช่วย
NULL	เป็นค่าพิเศษที่หมายถึง “การไม่มีค่าใดๆอยู่”

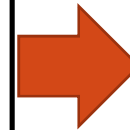
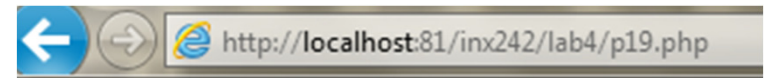
ฟังก์ชันเบื้องต้นของ PHP - gettype

- ฟังก์ชัน `gettype` เป็นคำสั่งตรวจสอบชนิดข้อมูลของตัวแปร โดยจะให้ผลลัพธ์เป็นชื่อชนิดข้อมูล

```
<?php
$a = 12345;
$b = 9.99;
$c = "Web Programming";
$d = TRUE;
$e=-12;

echo gettype($a)."<br />";
echo gettype($b)."<br />";
echo gettype($c)."<br />";
echo gettype($d)."<br />";
echo gettype($e)."<br /><br />";

$e=false;
echo gettype($e)."<br />"
?>
```



integer
double
string
boolean
integer

boolean

ฟังก์ชันเบื้องต้นของ PHP – integer, float

- การแปลงชนิดข้อมูลของตัวแปรโดยวิธีการ casting ด้วย integer และ float โดยจะให้ผลลัพธ์เปลี่ยนแปลงไปตามประเภทที่ถูกกำหนด

```
<?php  
$a = 12345;  
$b = 9.99;  
$c = $a + $b;  
  
echo (float)$c."<br />";  
echo (integer)$b."<br />";  
?>
```



```
12354.99  
9
```

ฟังก์ชันเบื้องต้นของ PHP - isset

- ฟังก์ชัน `isset` เป็นคำสั่งตรวจสอบการถูกกำหนดค่าหรือประกาศค่าของตัวแปร

```
<?php
if (isset($name)){
    print " มีการกำหนดค่าแล้ว คือ " . $name . "<hr />";
}
else{
    print " ยังไม่มีการกำหนดค่าตัวแปร " . "<hr />";
}
?>
```

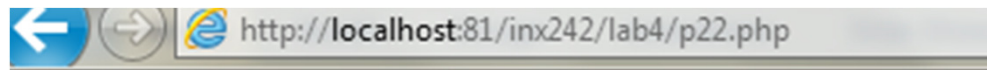
- หาก `$name=5` จะมีการแสดงผลทางหน้าจอ คือ **มีการกำหนดค่าแล้วคือ 5**
- หากไม่มีการกำหนด `$name` เลย จะมีการแสดงผลทางหน้าจอคือ **ยังไม่มีกำหนดค่าตัวแปร**

ฟังก์ชันเบื้องต้นของ PHP - unset

- ฟังก์ชัน unset เป็นคำสั่งยกเลิก / ทำลาย / ล้างค่า ตัวแปร
- ทรัพยากรระบบมีจำกัด ถ้ามีการใช้งานจำนวนมาก Bandwidth หน่วยความจำลดลง
- เมื่อไม่มีการใช้งานตัวแปรใดๆ ควรทำลายตัวแปรนั้นเพื่อคืนทรัพยากรให้กับระบบ

```
<?php
$a = 10;
echo "ค่า \$a ก่อน unset เท่ากับ ".$a."<br />";

unset($a);
echo "ค่า \$a หลัง unset เท่ากับ ".$a."<br />";
?>
```



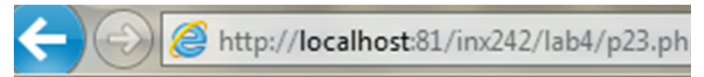
ค่า \$a ก่อน unset เท่ากับ 10
ค่า\$a หลัง unset เท่ากับ

ฟังก์ชันเบื้องต้นของ PHP - empty

- ฟังก์ชัน empty เป็นคำสั่งตรวจสอบค่าว่าง

```
<?php
$a = 10;
echo "ค่า \$a ก่อน unset เท่ากับ ".$a."<br />";
if (empty($a)){
    echo "ตัวแปร a ไม่มีค่า <br />";
}
else{
    echo "ตัวแปร \$a มีค่าเป็น $a <br />";
}
echo "<br /><br />";

unset($a);
echo "ค่า \$a หลัง unset เท่ากับ ".$a."<br />";
if (empty($a)){
    echo "ตัวแปร a ไม่มีค่า <br />";
}
else{
    echo "ตัวแปร a มีค่าเป็น $a <br />";
}
?>
```



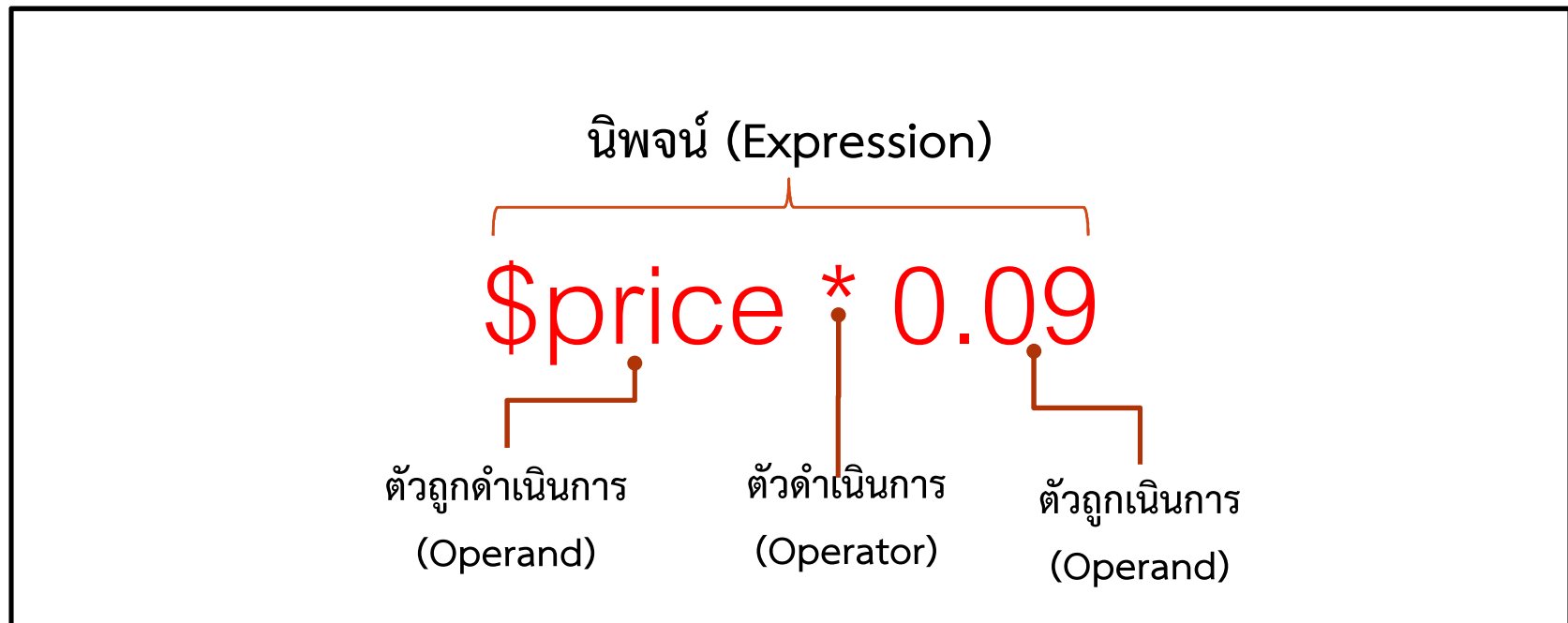
ค่า \$a ก่อน unset เท่ากับ 10
ตัวแปร \$a มีค่าเป็น 10



ค่า \$a หลัง unset เท่ากับ
ตัวแปร a ไม่มีค่า

นิพจน์ (Expression)

- นิพจน์ (Expression) หมายถึง ส่วนของโค้ดโปรแกรม**ที่มีค่า** ไม่ว่าจะค่าชนิดใดก็ตาม โดยสามารถสร้างนิพจน์ที่ซับซ้อนขึ้นโดยการนำตัวแปร หรือค่าคงที่มาดำเนินการ เช่น `$discount = $price * 0.90;`



ตัวดำเนินการ (Operator)

- ตัวดำเนินการ (Operator) คือ เครื่องหมายหรือสัญลักษณ์ที่ใช้ดำเนินการกับค่าใดๆ แล้วให้ผลลัพธ์เป็นค่าใหม่ ตัวดำเนินการในภาษา PHP แบ่งออกเป็นกลุ่มต่างๆ ได้แก่
 - ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)
 - ตัวดำเนินการสตริง (String Operator)
 - ตัวดำเนินการกำหนดค่า (Assignment Operator)
 - ตัวดำเนินการเพิ่ม/ลดค่า (Incrementing/Decrementing Operator)
 - ตัวดำเนินการเปรียบเทียบ (Comparison Operator)
 - ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator)
 - ตัวดำเนินการอาร์เรย์ (Array Operator)
 - ตัวดำเนินการระดับบิต (Bitwise Operator)
 - ตัวดำเนินการประมวลผล (Execution Operator)

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

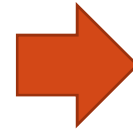
- ใช้คำนวณทางคณิตศาสตร์

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	ผลลัพธ์ (สมมติ \$a = 10, \$b = 4)	หมายเหตุ
+	บวก (Addition)	$\$a + \b	14	
-	ลบ (Subtraction)	$\$a - \b	6	
*	คูณ (Multiplication)	$\$a * \b	40	
/	หาร (Division)	$\$a / \b	2.5	ให้ค่าเป็น Float เสมอ
%	โมดูลัส (Modulus) คือ การหาเศษจากการหาร	$\$a \% \b	2	

ตัวดำเนินการทางสตริง (String Operator)

- ได้แก่ตัวดำเนินการ **.** (จุด หรือ dot) ที่ใช้เชื่อมต่อสตริง (String Connector)

```
<?php  
$name = "Alex";  
$surname = "Ferguson";  
$fullname = $name." ".$surname;  
echo $fullname;  
?>
```



Alex Ferguson

ตัวดำเนินการกำหนดค่า (Assignment Operator)

- โอเปอเรเตอร์สำหรับกำหนดค่าเป็นโอเปอเรเตอร์ที่มีการใช้อยู่ตลอดเวลา ได้แก่ เครื่องหมาย = แต่นอกจากนี้ยังมีเครื่องหมายอื่นๆ ที่ใช้สำหรับกำหนดค่าอีก คือ +=, -=, *=, /=, %= และ .= ซึ่งมีการดำเนินการดังนี้

ตัวดำเนินการ	ตัวอย่าง	ผลลัพธ์ (สมมติ \$a = 10, \$b = 4)
=	$\$c = \b	$\$c = 4$
+=	$\$a += \b	$\$a = \$a + \$b = 10 + 4 = 14$
-=	$\$a -= \b	$\$a = \$a - \$b = 10 - 4 = 6$
*=	$\$a *= \b	$\$a = \$a * \$b = 10 * 4 = 40$
/=	$\$a /= \b	$\$a = \$a / \$b = 10 / 4 = 2.5$
%=	$\$a \% = \b	$\$a = \$a \% \$b = 10 \% 4 = 2$
.=	$\$a .= \b	$\$a = 104$

ตัวดำเนินการเพิ่ม/ลดค่า (Incrementing/Decrementing Operator)

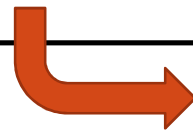
- โอเปอเรเตอร์สำหรับเพิ่มและลดค่าใน PHP มีทั้งแบบใช้หน้าตัวแปรและ หลังตัวแปร เช่นเดียวกับภาษา C ซึ่งมีรูปแบบดังตาราง

ตัวดำเนินการ	ชื่อ	ความหมาย
$\$a++$	Pre-increment	<ul style="list-style-type: none">■ ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการมีค่าคงเดิม แต่หลังจากเรียกใช้งานตัวดำเนินการ ให้เพิ่มค่าขึ้น 1 ค่า■ ส่งค่าตัวแปร $\\$a$ แล้วจึงเพิ่มค่าตัวแปร $\\$a$ ขึ้นอีก 1
$++\$a$	Post-increment	<ul style="list-style-type: none">■ ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการเพิ่มค่าขึ้น 1 ค่า■ เพิ่มค่าตัวแปร $\\$a$ ขึ้น 1 แล้วจึงคืนค่าตัวแปร $\\$a$
$\$a--$	Pre-decrement	<ul style="list-style-type: none">■ ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการมีค่าคงเดิม แต่หลังจากเรียกใช้งานตัวดำเนินการ ให้ลดค่าลง 1 ค่า■ ส่งค่าตัวแปร $\\$a$ แล้วจึงลดค่าตัวแปร $\\$a$ ลงอีก 1
$--\$a$	Post-decrement	<ul style="list-style-type: none">■ ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการลดค่าลง 1 ค่า■ ลดค่าตัวแปร $\\$a$ ลง 1 แล้วจึงคืนค่าตัวแปร $\\$a$

ตัวดำเนินการเพิ่ม/ลดค่า (Incrementing/Decrementing Operator)

```
<?php
$a = 5;
$b = ++$a * 20;
echo "การดำเนินการที่ 1 : \$a = $a, \$b = $b".<br />";

$a = 5;
$b = $a++ * 20;
echo "การดำเนินการที่ 2 : \$a = $a, \$b = $b".<br />";
?>
```

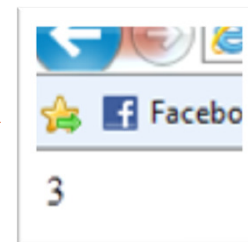


การดำเนินการที่ 1 : \$a = 6, \$b = 120
การดำเนินการที่ 2 : \$a = 6, \$b = 100

```
<?php
$c = 1;
$d = $c++ * ++$c; /* เท่ากับ 1 คูณกับ 3 */
echo $d;
?>
```



ประมวลผล



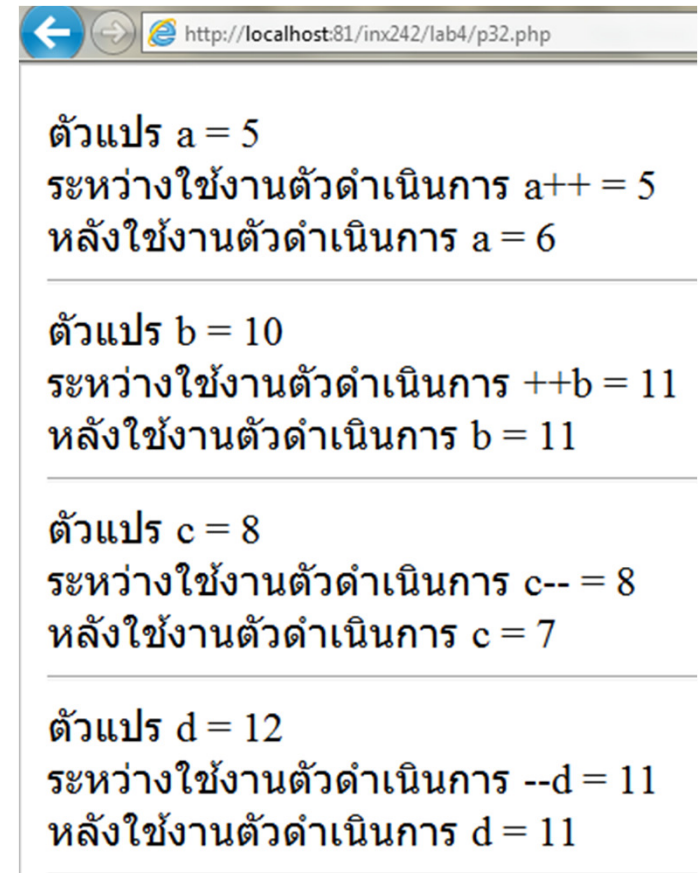
ตัวดำเนินการเพิ่ม/ลดค่า (Incrementing/Decrementing Operator)

```
<?php
$a = 5;
print "ตัวแปร a = " . $a . "<br>";
print "ระหว่างใช้งานตัวดำเนินการ a++ = " . $a++ . "<br>";
print "หลังใช้งานตัวดำเนินการ a = " . $a . "<hr>";

$b = 10;
print "ตัวแปร b = " . $b . "<br>";
print "ระหว่างใช้งานตัวดำเนินการ ++b = " . ++$b . "<br>";
print "หลังใช้งานตัวดำเนินการ b = " . $b . "<hr>";

$c = 8;
print "ตัวแปร c = " . $c . "<br>";
print "ระหว่างใช้งานตัวดำเนินการ c-- = " . $c-- . "<br>";
print "หลังใช้งานตัวดำเนินการ c = " . $c . "<hr>";

$d = 12;
print "ตัวแปร d = " . $d . "<br>";
print "ระหว่างใช้งานตัวดำเนินการ --d = " . --$d . "<br>";
print "หลังใช้งานตัวดำเนินการ d = " . $d . "<hr>";
?>
```



```
← → http://localhost:81/inx242/lab4/p32.php

ตัวแปร a = 5
ระหว่างใช้งานตัวดำเนินการ a++ = 5
หลังใช้งานตัวดำเนินการ a = 6

-----

ตัวแปร b = 10
ระหว่างใช้งานตัวดำเนินการ ++b = 11
หลังใช้งานตัวดำเนินการ b = 11

-----

ตัวแปร c = 8
ระหว่างใช้งานตัวดำเนินการ c-- = 8
หลังใช้งานตัวดำเนินการ c = 7

-----

ตัวแปร d = 12
ระหว่างใช้งานตัวดำเนินการ --d = 11
หลังใช้งานตัวดำเนินการ d = 11
```

ตัวดำเนินการเปรียบเทียบ (Comparison Operator)

- ใช้เปรียบเทียบค่า 2 ค่าว่าเท่ากัน ไม่เท่ากัน น้อยกว่า มากกว่า ฯลฯ หรือไม่ โดยจะให้ผลลัพธ์เป็นค่าตรรกะ ดังนั้นไม่ว่าตัวถูกดำเนินการทั้งสองจะเป็นค่าอะไรก็ตาม เมื่อใช้ตัวดำเนินการในกลุ่มนี้จะได้ผลลัพธ์เป็น TRUE หรือ FALSE ใดๆอย่างหนึ่งเสมอ ไม่มีทางเป็นค่าอื่นได้

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	ผลลัพธ์ (\$a = 10, \$b = 10.00, \$c = 5)
==	เท่ากับ (Equal)	\$a == \$b	TRUE
===	เหมือนกัน (Identical)	\$a === \$b	FALSE
!= หรือ <>	ไม่เท่ากับ (Not equal)	\$a != \$c	TRUE
!==	ไม่เหมือนกัน (Not identical)	\$a !== \$b	TRUE
<	น้อยกว่า (Less than)	\$a < \$b	FALSE
>	มากกว่า (Greater than)	\$a > \$b	FALSE
<=	น้อยกว่าหรือเท่ากับ (Less than or equal to)	\$a <= \$b	TRUE
>=	มากกว่าหรือเท่ากับ (Greater than or equal to)	\$a >= \$c	TRUE

ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator)

- โอเปอเรเตอร์ในกลุ่มนี้มักถูกเรียกใช้อยู่เสมอ โดยเฉพาะโอเปอเรเตอร์ and, or และ ! แต่นอกจากทั้งสามตัวนี้แล้วยังมี xor ซึ่งมีความหมายและวิธีใช้งานดังตาราง

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	ผลลัพธ์
&& หรือ and	และ (And)	$\$a \ \&\& \ \b	ให้ค่า TRUE เมื่อ $\$a$ และ $\$b$ เป็น TRUE ทั้งคู่ถ้าตัวใดตัวหนึ่งหรือทั้งสองตัวเป็น FALSE จะให้ค่า FALSE
หรือ or	หรือ (Or)	$\$a \ \ \b	ให้ค่า TRUE เมื่อตัวใดตัวหนึ่งหรือทั้งสองตัวเป็น TRUE แต่ถ้าเป็น FALSE ทั้งคู่จะให้ค่าเป็น FALSE
xor	Exclusive or	$\$a \ xor \ \c	ให้ค่า TRUE เมื่อ $\$a$ หรือ $\$b$ ตัวใดตัวหนึ่งเท่านั้นที่เป็น TRUE แต่ถ้าเป็น TRUE ทั้งคู่หรือเป็น FALSE ทั้งคู่จะให้ค่า FALSE
!	นิเสธ (Not)	$\$a \ != \ \b	ให้ค่าตรรกะที่ตรงกันข้ามกับค่าของ $\$a$ นั่นคือให้ค่า TRUE เมื่อ $\$a$ เป็น FALSE และให้ค่า FALSE เมื่อ $\$a$ เป็น TRUE

ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator)

```
<?php
$a = 10;
$b = 10.0;
$c = 100;
$result1 = ($a == $b) || ($a === $b); // $result1 จะมีค่าเป็น TRUE
$result2 = ($c > $a) && ($a > $b); // $result2 จะมีค่าเป็น FALSE
$result3 = ($c > $a) xor ($c > $b); // $result3 จะมีค่าเป็น FALSE
$result4 = !($a >= $b); // $result3 จะมีค่าเป็น FALSE
?>
```

สตริงในภาษา PHP

- สตริง (String) คือ กลุ่มของอักขระ (Character) ซึ่งอาจจะเป็นตัวอักษร ตัวเลข เครื่องหมายวรรคตอน หรือแม้แต่สัญลักษณ์ต่างๆ
- การเขียนค่าสตริงสามารถแบ่งได้เป็น 3 รูปแบบ คือ
 - สตริงแบบ Single quote
 - สตริงแบบ Double quote
 - สตริงแบบ Heredoc

สตริงในภาษา PHP – Single quote

- เป็นรูปแบบที่ใช้เครื่องหมายคำพูดขีดเดี่ยว (') ครอบไว้ เช่น

```
<?php  
    echo 'This is a book.';  
?>
```

- ถ้าต้องการให้ผลลัพธ์ที่แสดงออกมานั้นมีเครื่องหมาย ' ด้วย ให้ใช้เครื่องหมาย backslash (\) กำกับไว้ข้างหน้า

```
<?php  
    echo 'I\'ll always love you.';  
?>
```

สตริงในภาษา PHP – Double quote

- เป็นรูปแบบที่ใช้เครื่องหมายคำพูดสองขีด (") ครอบไว้ เช่น

```
<?php  
echo Single quote "example";  
?>
```

- PHP จะตีความเครื่องหมาย backslash ภายในค่าสตริงดังรายละเอียดในตาราง

สัญลักษณ์ (Escape Character)	ความหมาย
\n	การขึ้นบรรทัดใหม่ (New line)
\r	การปิดแคร่ – เริ่มต้นที่คอลัมน์แรกของบรรทัด (Carriage Return)
\t	การเว้นระยะแท็บในแนวนอน (แสดง Tab)
\\	แสดงเครื่องหมาย backslash 1 ตัว
\\$	แสดงเครื่องหมายดอลลาร์
\"	แสดงเครื่องหมายคำพูดสองขีด

สตริงในภาษา PHP – Heredoc

- สตริงแบบ heredoc จะเริ่มต้นด้วย <<< ตามด้วย identifier ซึ่งเป็นค่าอะไรก็ได้ จากนั้นขึ้นบรรทัดใหม่ แล้วระบุสตริงที่ต้องการลงไป (สามารถขึ้นบรรทัดใหม่ภายในค่าสตริงได้) เสร็จแล้วให้ขึ้นบรรทัดใหม่แล้วปิดท้ายด้วย identifier คำเดิม
- รูปแบบมาตรฐานของสตริงแบบ heredoc

```
<<<identifier  
สตริง  
identifier
```

- สตริงแบบ heredoc จะตีความเครื่องหมาย backslash ที่อยู่ในสตริงเช่นเดียวกับสตริงแบบ double quote (เหมือนตารางในสไลด์ก่อนหน้า) ยกเว้นเครื่องหมาย double quote ซึ่งสามารถใส่เข้าไปได้เลยโดยไม่ต้องมีเครื่องหมาย backslash นำหน้า

สตริงในภาษา PHP – Heredoc

```
<?php  
$str = <<<heredoc  
สวัสดีครับ  
heredoc;  
echo $str;  
?>
```



ประมวลผล

สวัสดีครับ

ฟังก์ชันสตริงในภาษา PHP – echo, print

- คำสั่งหรือฟังก์ชันทั้งสองนี้ใช้ในการแสดงข้อความที่หน้าเว็บเพจ หรือใช้ส่งแท็กในภาษา HTML ไปที่ Web Browser การพิมพ์สามารถใช้วงเล็บเหมือนการใช้ฟังก์ชันหรือไม่ก็ได้

```
<?php
echo ("HTML<br>\n");           //แบบใช้วงเล็บ
echo "Javascript<br>\n";       //แบบไม่ใช้วงเล็บ
print ("ASP<br>\n");           //แบบใช้วงเล็บ
print "PHP";                   //แบบไม่ใช้วงเล็บ
?>
```



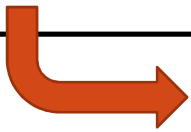
```
HTML
Javascript
ASP
PHP
```

ฟังก์ชันสตริงในภาษา PHP – echo, print

- การเขียน PHP ยังคงต้องอาศัย Tag ของภาษา HTML เพื่อการรับและแสดงผลข้อมูลเสมอ ไม่ว่าจะเป็นการรับข้อมูลด้วยฟอร์มที่สร้างจากแท็ก <form>, การแสดงรูปภาพด้วยแท็ก หรือการสร้างตารางด้วยแท็ก <table> ซึ่งในขณะที่เขียนสคริปต์ PHP อยู่และต้องการสร้างแท็กต่างๆในภาษา HTML เพื่อส่งไปยัง Web Browser สามารถทำได้ทันทีโดยใช้ echo หรือ print ดังตัวอย่าง

```
<?php
echo "<h3>Materials requirement</h3>";
echo "<table border=3>";
echo "<tr><th rowspan=2>Item no. <th colspan=3>Month of";
echo "<tr><th>Jan<th>Feb<th>Mar";
echo "<tr><td>A<td align=right>100<td align=right>98<td align=right>120";
echo "<tr><td>B<td align=right>500<td align=right>460<td align=right>680";
echo "<tr><td>A<td align=right>1080<td align=right>875<td align=right>1214";
echo "<tr><td>A<td align=right>14<td align=right>9<td align=right>13";
echo "</table>";
?>
```

Materials requirement



Item no.	Month of		
	Jan	Feb	Mar
A	100	98	120
B	500	460	680
A	1080	875	1214
A	14	9	13

ฟังก์ชันสตริงในภาษา PHP – การเชื่อมต่อข้อความ

- การเชื่อมต่อข้อความหรือตัวแปรชนิดสตริงเข้าด้วยกันนั้นจะใช้ตัวดำเนินการ (operator) . “จุด (Dot)”

```
<?php  
$name = "Alex";  
$surname = "Ferguson";  
$fullname = $name." ".$surname;  
echo $fullname;  
?>
```



ประมวลผล

Alex Ferguson

ฟังก์ชันสตริงในภาษา PHP – ตัดข้อความ (substr)

- ฟังก์ชัน substr เป็นฟังก์ชันที่มีการใช้งานบ่อยฟังก์ชันหนึ่ง ใช้สำหรับตัดตัวอักษรหรือข้อความออกมาจากข้อความเดิม
- รูปแบบมาตรฐานของฟังก์ชัน substr

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

substr(ข้อความ , ตำแหน่งเริ่มต้นของคำที่ต้องการเหลือไว้, จำนวนตัวอักษรที่ต้องการเหลือไว้)

- string str ข้อความที่ต้องการนำมาตัดตัวอักษร
- int start ตำแหน่งเริ่มต้นของตัวอักษรที่ต้องการตัด มีค่าเป็นบวกหรือลบก็ได้ โดยตัวอักษรตัวแรกเริ่มที่ค่า 0
- int length จำนวนตัวอักษรที่ต้องการตัด มีค่าเป็นบวกหรือลบก็ได้ และจะระบุหรือไม่ก็ได้หากระบุไว้เป็นค่าบวก และทำการตัดตั้งแต่ตำแหน่ง start ไปเป็นจำนวนเท่ากับ length แต่หากไม่ระบุจะทำการตัดตั้งแต่ตำแหน่ง start ไปจนสุดข้อความ

การตัดข้อความเป็นลักษณะการตัดข้อความแบบเก็บ/เหลือไว้

ฟังก์ชันสตริงในภาษา PHP – ตัดข้อความ (substr)

```
<?php  
echo substr("Hello world!",6);  
?>
```



ประมวลผล

world!

```
<?php  
echo substr("Hello world!",6,5);  
?>
```



ประมวลผล

world

ฟังก์ชันสตริงในภาษา PHP – ตัดข้อความ (substr)

- หากค่าตำแหน่งเริ่มต้น (start) เกินกว่าความยาวของข้อความลบด้วย 1 (ตัวอักษรตัวแรกนับเป็นตำแหน่งที่ 0) ฟังก์ชัน substr จะให้ค่าเป็นเท็จ (False)

```
<?php
$text = "ABCDEFGHJKLMNOP";
$newstr = substr($text,41, 4);
if ($newstr) {
    print $newstr;
}
else {
    print "The message is shorter than the start.";
}
?>
```



The message is shorter than the start.

ฟังก์ชันสตริงในภาษา PHP – ตัดข้อความ (substr)

```
<?php
$a="I love you";
echo substr($a,2,4)."<br />";
echo substr($a,0,6)."<br />";
?>
```

```
<?php
$b="Loving you";
echo substr($b,-3,3)."<br />";
echo substr($b,0,-4)."<br />";
?>
```

ตำแหน่ง	0	1	2	3	4	5	6	7	8	9
ข้อความ	I		l	o	v	e		y	o	u

ต้องการตัดให้เหลือคำว่า love

- ตำแหน่งเริ่มต้นของคำที่ต้องการเหลือไว้ คือที่ 2
- จำนวนตัวอักษรที่ต้องการให้เหลือไว้คือ 4 ตัวอักษร

ต้องการตัดให้เหลือคำว่า I love

- ตำแหน่งเริ่มต้นของคำที่ต้องการเหลือไว้ คือที่ 0
- จำนวนตัวอักษรที่ต้องการให้เหลือไว้คือ 6 ตัวอักษร

ตำแหน่ง	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
ข้อความ	L	o	v	i	n	g		y	o	u

ต้องการตัดให้เหลือคำว่า you

- ตำแหน่งเริ่มต้นของคำที่ต้องการเหลือไว้ คือที่ -3
- จำนวนตัวอักษรที่ต้องการให้เหลือไว้คือ 3 ตัวอักษร

ต้องการตัดให้เหลือคำว่า Loving

- ตำแหน่งเริ่มต้นของคำที่ต้องการเหลือไว้ คือตำแหน่งที่ 0
- จำนวนตัวอักษรที่ต้องการให้เหลือไว้คือ -4 ตัวอักษร

ฟังก์ชันสตริงในภาษา PHP – แทรก/แทนที่ (substr_replace)

- หากต้องการแทรกหรือแทนที่ตัวอักษร จะใช้ฟังก์ชัน substr_replace
- รูปแบบมาตรฐานของฟังก์ชัน substr_replace

substr_replace(string str , string replace, int start, [int length])

- string str ข้อความที่ต้องการนำตัวอักษรหรือข้อความอื่นมาแทรกหรือแทนที่ตัวอักษรข้อความหรือตัวอักษรที่นำมาแทรกหรือแทนที่
- string replace ข้อความหรือตัวอักษรที่นำมาแทรกหรือแทนที่
- int start ตำแหน่งเริ่มต้นที่ต้องการแทรกหรือแทนที่ มีค่าเป็นบวกหรือลบก็ได้ โดยตัวอักษรตัวแรกเริ่มที่ค่า 0
- int length จำนวนตัวอักษรที่ต้องการแทนที่ มีค่าเป็นบวกหรือลบก็ได้ และจะระบุหรือไม่ก็ได้ หากระบุไว้เป็นค่าบวก จะทำการแทนที่ตั้งแต่ตำแหน่ง start ไปเป็นจำนวนเท่ากับ length แต่หากไม่ระบุ ตัวอักษรตั้งแต่ตำแหน่ง start ไปจนสุด ข้อความจะถูกแทนที่ด้วยค่าใน string replace

```
<?php  
echo substr_replace("Hello world","earth",6);  
?>
```



Hello earth

ฟังก์ชันสตริงในภาษา PHP – แทรก/แทนที่ (substr_replace)

- ในกรณีที่ต้องการแทนที่ตัวอักษรหรือคำในข้อความด้วยตัวอักษรหรือคำอื่น สามารถใช้ฟังก์ชัน `str_replace` ได้
- รูปแบบมาตรฐานของฟังก์ชัน `str_replace`

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

- `mixed search` คำที่ต้องการให้ถูกแทนที่
- `mixed replace` คำที่จะนำมาแทนที่
- `mixed subject` ประโยคหรือข้อความ
- `int & count` ตัวแปรสำหรับรับค่าจำนวนคำหรือส่วนของข้อความที่ถูกแทนที่ซึ่งจะถูกส่งผ่านโดยการอ้างอิง (passe by reference) [เริ่มมีใน PHP5]

หมายเหตุ : ฟังก์ชัน `str_replace` พิจารณาความแตกต่างระหว่างอักษรเล็กกับใหญ่ กรณีที่ไม่สนใจให้ใช้ฟังก์ชัน `str_ireplace`

ฟังก์ชันสตริงในภาษา PHP – แทรก/แทนที่ (substr_replace)

ตัวอย่างการใช้ฟังก์ชัน str_replace

```
<?php
    $error = "Love me love my frog";
    $correct = str_replace("frog", "dog", $error, $count);
    echo "บรรทัดที่ 1 : ".$correct." (มีการแทนที่ ".$count." ครั้ง)";
?>
```



บรรทัดที่ 1 : Love me love my dog (มีการแทนที่ 1 ครั้ง)

ฟังก์ชันการแทนที่ค่าหรือตัวอักษร - str_replace

str_replace(ข้อความเดิม , คำที่จะถูกแทนที่, คำใหม่ที่แทนที่)

```
<?php  
$webcamt="www.camt.cmu.ac.th";  
echo $webcamt."<br />";  
echo str_replace("ac.th","org",$webcamt);  
?>
```



www.camt.cmu.ac.th
www.camt.cmu.org

ฟังก์ชันการแทนที่คำหรือตัวอักษร – str_replace

- ฟังก์ชัน str_replace() สามารถคืนค่าให้ได้ทั้งสตริงและอาร์เรย์ ส่วนอาร์กิวเมนต์ search, replace และ subject ก็สามารเป็นได้ทั้งสตริงและอาร์เรย์เช่นกัน
- หากต้องการให้มีการแทนที่คำมากกว่า 1 คำ สามารถกำหนดให้ search และ replace เป็นตัวแปรอาร์เรย์ได้ และหากมีข้อความมากกว่า 1
- ข้อความที่ต้องการให้มีการแทนที่คำ สามารถกำหนดให้ subject เป็นตัวแปรอาร์เรย์ของข้อความหลายๆข้อความได้ ซึ่งจะทำให้ฟังก์ชัน str_replace คืนค่ากลับมาเป็นอาร์เรย์เช่นกัน

ฟังก์ชันการแทนที่ค่าหรือตัวอักษร – str_replace

ตัวอย่างการใช้ฟังก์ชัน str_replace กับรูปแบบอาร์เรย์

```
<?php
$curriculums = "CAMT consists of MMIT SE ANI and KM curriculum";
$code = array("CAMT", "MMIT", "SE", "ANI", "KM");
$fullname = array("College of Arts, Media and Technology", "Modern
Management and Information Technology", "Software Engineer", "Animation",
"Knowledge Management");
$newmessage = str_replace($code, $fullname, $curriculums, $count);
echo "บรรทัดที่ 1 : ".$newmessage." (มีการแทนที่ทั้งหมด ".$count." ครั้ง)";
?>
```



บรรทัดที่ 1 : College of Arts, Media and Technology consists of Modern Management and Information Technology Software Engineer Animation and Knowledge Management curriculum (มีการแทนที่ทั้งหมด 5 ครั้ง)

หมายเหตุ : ลักษณะข้อความที่ได้เป็นเช่นนี้ แต่
ไม่มีการขึ้นบรรทัดใหม่

ฟังก์ชันการหาตำแหน่งของค่าหรือตัวอักษร - strpos

- ฟังก์ชันที่ช่วยให้การหาตำแหน่งของตัวอักษรหรือค่าในข้อความทำได้ง่ายขึ้น เพียงแต่ใส่อาร์กิวเมนต์ที่เป็นข้อความที่ต้องการตรวจสอบ และตัวอักษรหรือค่าที่ต้องการค้นหาลงไปเท่านั้น
- รูปแบบมาตรฐานของฟังก์ชัน strpos

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

- string haystack ข้อความที่ต้องการตรวจสอบ
- string needle ข้อความหรือตัวอักษรที่ต้องการค้นหา
- int offset ตำแหน่งที่ให้เริ่มค้นหา (อักขรตัวแรกนับเป็น 0) เป็น optional parameter คือจะระบุหรือไม่ระบุก็ได้

หมายเหตุ :

1. ฟังก์ชัน strpos ใช้หาตำแหน่งสตริงที่ปรากฏครั้งแรก กรณีต้องการหาตำแหน่งสุดท้ายให้ใช้ฟังก์ชัน strrpos
2. ส่วนกรณีฟังก์ชัน stripos และ strpos นั้นมีหน้าที่การทำงานคล้ายคลึงกับ strpos และ strrpos ตามลำดับ แตกต่างตรงที่ฟังก์ชัน stripos และ strpos ไม่พิจารณาความแตกต่างระหว่างอักขรเล็กกับอักขรใหญ่

ฟังก์ชันการหาตำแหน่งของค่าหรือตัวอักษร - strpos

- ตัวอย่างการใช้ฟังก์ชัน strpos

```
<?php
    $number = "1234567890";
    $find5 = strpos($number, "5");
    echo $find5;
?>
```



4

```
<?php
$emailaddress = "INX242@camt.info";
$name = substr($emailaddress, 0, strpos($emailaddress, "@"));
$domain = substr($emailaddress, strpos($emailaddress, "@")+1);
echo "Name = $name<br>";
echo "Domain = $domain";
?>
```



Name = INX242
Domain = camt.info

ฟังก์ชันการหาความยาวของข้อความ – strlen

- เป็นฟังก์ชันใช้หาจำนวนตัวอักษรทั้งหมดในข้อความ
- รูปแบบมาตรฐานของฟังก์ชัน strlen **strlen(string)**
 - string ข้อความที่ต้องการตรวจสอบความยาว
- ตัวอย่างการใช้ฟังก์ชัน strlen

```
<?php  
$message = "1234 6789";  
$length = strlen($message);  
echo "$message มีความยาวทั้งสิ้น ".$length." คำ";  
?>
```



1234 6789 มีความยาวทั้งสิ้น 9 คำ

การแปลง Case ของตัวอักษรในสตริง

- การแปลงข้อความเป็นตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กสามารถทำได้ 4 ลักษณะ โดยใช้ฟังก์ชันต่างๆ ต่อไปนี้

ฟังก์ชัน	คำอธิบาย
strtoupper	แปลงตัวอักษรทั้งหมดเป็นตัวอักษรใหญ่ (Upper case)
strtolower	แปลงตัวอักษรทั้งหมดเป็นตัวอักษรเล็ก (Lower case)
ucfirst	แปลงตัวอักษรแรกของสตริงเป็นตัวอักษรใหญ่
ucwords	แปลงตัวอักษรแรกของแต่ละคำในสตริงเป็นตัวอักษรใหญ่

ฟังก์ชันเกี่ยวกับสตริงอื่นๆ

ฟังก์ชันที่ควรศึกษาเพิ่มเติม

- ฟังก์ชันการลบ Whitespace : ฟังก์ชัน trim (ตัดช่องว่างทั้งด้านหน้าและหลังของข้อความทั้งหมด), ltrim trim (ตัดช่องว่างด้านหน้า/ซ้ายของข้อความทั้งหมด), rtrim (ตัดช่องว่างด้านหลัง/ขวาของข้อความทั้งหมด), chop
- ฟังก์ชันการรวมค่าในอาร์เรย์ให้เป็นสตริง : ฟังก์ชัน implode
- ฟังก์ชันการแยกค่าในอาร์เรย์ให้เป็นสตริง : ฟังก์ชัน explode
- ฟังก์ชันการแปลงอักขระที่มีความหมายพิเศษในภาษา HTML ไปเป็นสัญลักษณ์ในภาษา HTML : ฟังก์ชัน htmlspecialchars

ฟังก์ชันเกี่ยวกับเลขจำนวน

- ภาษา PHP มีชนิดข้อมูลแบบเลขจำนวนอยู่ 2 ชนิด ได้แก่ integer และ float (หรือ double)
- ในการสร้างตัวแปร ไม่จำเป็นต้องกำหนดชนิด (Type) ให้กับตัวแปร เนื่องจาก PHP จะกำหนดชนิดของตัวแปรให้โดยอัตโนมัติตามชนิดของข้อมูลที่บรรจุเข้าไป เช่นในประโยค `$x = 5.13;` ตัวแปร `$x` จะถูกกำหนดเป็นชนิด double หรือในประโยค `$x = 5;` ตัวแปร `$x` จะถูกกำหนดให้เป็น integer นอกจากนี้ในประโยค `$x = "5.13";` ตัวแปร `$x` จะถูกกำหนดให้เป็นชนิดสตริง จะเห็นว่าความยืดหยุ่นในการจัดการข้อมูลของ PHP อาจสร้างให้เกิดความสับสน
- ดังนั้นจึงควรต้องทราบเกี่ยวกับการจัดการข้อมูลที่เป็นตัวเลขคือ การตรวจสอบชนิดของข้อมูลหรือตัวแปร

ฟังก์ชันตรวจสอบชนิดของข้อมูลหรือตัวแปร - gettype

- ฟังก์ชัน `gettype` เป็นคำสั่งตรวจสอบชนิดข้อมูลของตัวแปร โดยจะให้ผลลัพธ์เป็นชื่อชนิดข้อมูล

```
<?php
$a = 455;
$b = 3.1745;
$c = "10.10";
$z = $a + $b + $c;
print "\$a = $a is " .
gettype($a) . "<br>";
print "\$b = $b is " .
gettype($b) . "<br>";
print "\$c = $c is " .
gettype($c) . "<br>";
print "\$z = $z is " .
gettype($z) . "<br>";
?>
```



```
$a = 455 is integer
$b = 3.1745 is double
$c = 10.10 is string
$z = 468.2745 is double
```

ฟังก์ชันตรวจสอบชนิดของข้อมูลหรือตัวแปร – is_int

- ฟังก์ชัน is_int เป็นคำสั่งตรวจสอบชนิดข้อมูลว่าเป็นชนิด integer หรือไม่ โดยจะให้ค่า TRUE ถ้าใช่และให้ค่า FALSE ถ้าไม่ใช่
- รูปแบบมาตรฐานของฟังก์ชัน is_int **is_int(variable)**
 - variable ตัวแปรที่ต้องการตรวจสอบค่า

```
<?php
$var1 = "8247";
$var2 = 76;

if(is_int($var1))
    echo "\$var1 คือตัวแปรชนิด integrer"."<br>";
else
    echo "\$var1 ไม่ใช่ตัวแปรชนิด integrer"."<br>";

if(is_int($var2))
    echo "\$var2 คือตัวแปรชนิด integrer"."<br>";
else
    echo "\$var2 ไม่ใช่ตัวแปรชนิด integrer"."<br>";
?>
```

**\$var1 ไม่ใช่ตัวแปรชนิด integer
\$var2 คือตัวแปรชนิด integer**

ฟังก์ชันตรวจสอบชนิดของข้อมูลหรือตัวแปร – is_float

- ฟังก์ชัน is_float เป็นคำสั่งตรวจสอบชนิดข้อมูลว่าเป็นชนิด float หรือไม่ โดยจะให้ค่า TRUE ถ้าใช่และให้ค่า FALSE ถ้าไม่ใช่
- รูปแบบมาตรฐานของฟังก์ชัน is_float **is_float(variable)**
 - ตัวแปรที่ต้องการตรวจสอบค่า

```
<?php
$var1 = "8247";
$var2 = 76;

if(is_float($var1))
    echo "\$var1 คือตัวแปรชนิด float". "<br>";
else
    echo "\$var1 ไม่ใช่ตัวแปรชนิด float". "<br>";

if(is_float($var2))
    echo "\$var2 คือตัวแปรชนิด float". "<br>";
else
    echo "\$var2 ไม่ใช่ตัวแปรชนิด float". "<br>";

?>
```



**\$var1 ไม่ใช่ตัวแปรชนิด float
\$var2 ไม่ใช่ตัวแปรชนิด float**

ฟังก์ชันตรวจสอบชนิดของข้อมูลหรือตัวแปร – is_numeric

- ฟังก์ชัน is_numeric เป็นคำสั่งตรวจสอบชนิดข้อมูลว่าเป็นชนิดเลขจำนวนหรือเป็นสตริงตัวเลข (Numeric string) หรือไม่ โดยจะให้ค่า TRUE ถ้าใช่และให้ค่า FALSE ถ้าไม่ใช่
- รูปแบบมาตรฐานของฟังก์ชัน is_numeric **is_numeric(variable)**
 - variable ตัวแปรที่ต้องการตรวจสอบค่า

```
<?php
$var1 = "8247";
$var2 = 76;
if(is_numeric($var1))
    echo "\$var1 คือเลขจำนวนหรือสตริงที่มีรูปแบบเป็นเลขจำนวน."<br>;
else
    echo "\$var1 ไม่ใช่เลขจำนวนหรือสตริงที่มีรูปแบบเป็นเลขจำนวน."<br>;

if(is_numeric($var2))
    echo "\$var2 คือเลขจำนวนหรือสตริงที่มีรูปแบบเป็นเลขจำนวน."<br>;
else
    echo "\$var2 ไม่ใช่เลขจำนวนหรือสตริงที่มีรูปแบบเป็นเลขจำนวน."<br>;
?>
```



\$var1 คือเลขจำนวนหรือสตริงที่มีรูปแบบเป็นเลขจำนวน
\$var2 คือเลขจำนวนหรือสตริงที่มีรูปแบบเป็นเลขจำนวน

ฟังก์ชันการปัดเศษทศนิยม แบบปัดเศษขึ้น – ceil

- ฟังก์ชันที่ให้ค่าเป็นเลขจำนวนเต็มโดยการปัดเศษขึ้น ถึงแม้ว่าผลลัพธ์จะเป็นจำนวนเต็ม แต่ชนิดข้อมูลยังเป็น float
- รูปแบบมาตรฐานของฟังก์ชัน ceil **ceil(float)**
 - float เป็นเลขจำนวนแบบ float

```
<?php
echo(ceil(0.60) . "<br />");
echo(ceil(0.40) . "<br />");
echo(ceil(5) . "<br />");
echo(ceil(5.1) . "<br />");
echo(ceil(-5.1) . "<br />");
echo(ceil(-5.9))
?>
```



```
บรรทัดที่ 1 : 1
บรรทัดที่ 2 : 1
บรรทัดที่ 3 : 5
บรรทัดที่ 4 : 6
บรรทัดที่ 5 : -5
บรรทัดที่ 6 : -5
```

ฟังก์ชันการปัดเศษทศนิยม แบบปัดเศษลง – floor

- ฟังก์ชันที่ให้ค่าเป็นเลขจำนวนเต็มโดยการปัดเศษลง ถึงแม้ว่าผลลัพธ์จะเป็นจำนวนเต็ม แต่ชนิดข้อมูลยังเป็น float
- รูปแบบมาตรฐานของฟังก์ชัน floor
 - float เป็นเลขจำนวนแบบ float

floor(float)

```
<?php
echo(floor(0.60) . "<br />");
echo(floor(0.40) . "<br />");
echo(floor(5) . "<br />");
echo(floor(5.1) . "<br />");
echo(floor(-5.1) . "<br />");
echo(floor(-5.9))
?
```



```
บรรทัดที่ 1 : 0
บรรทัดที่ 2 : 0
บรรทัดที่ 3 : 5
บรรทัดที่ 4 : 5
บรรทัดที่ 5 : -6
บรรทัดที่ 6 : -6
```

ฟังก์ชันที่ใช้ประมาณค่า/ปัดเศษ – round

- ฟังก์ชันที่ใช้หาค่าประมาณของเลขจำนวนโดยให้ความแม่นยำตามที่กำหนด โดยความแม่นยำหมายถึงตำแหน่งของตัวเลขหลังจุดทศนิยม กรณีที่กำหนดความแม่นยำเป็นค่าบวก นอกจากนี้สามารถกำหนดความแม่นยำเป็นค่าศูนย์หรือค่าลบได้ด้วย (ค่าปกติคือศูนย์) ซึ่งจะหมายถึงตำแหน่งของตัวเลขข้างหน้าจุดทศนิยม (0 = หลักหน่วย. -1 = หลักสิบ ฯลฯ)
- รูปแบบมาตรฐานของฟังก์ชัน round

round(float [, int precision])

- float เป็นเลขจำนวนแบบ float
- int precision ความแม่นยำของผลลัพธ์ที่ต้องการ

ฟังก์ชันที่ใช้ประมาณค่า/ปัดเศษ – round

- ตัวอย่างการใช้ฟังก์ชัน round

```
<?php  
echo(round(0.60) . "<br />");  
echo(round(0.50) . "<br />");  
echo(round(0.49) . "<br />");  
echo(round(-4.40) . "<br />");  
echo(round(-4.60))  
?>
```



```
1  
1  
0  
-4  
-5
```

ฟังก์ชันเกี่ยวกับวันที่และเวลา

- เว็บแอปพลิเคชันจะต้องเกี่ยวข้องกับวัน (Date) และ เวลา (Time) อยู่เสมอ เช่น ในระบบซื้อ-ขาย สินค้าจะมีการเก็บวัน/เวลาที่ลูกค้าสั่งซื้อสินค้าเข้ามา ในระบบเว็บบอร์ดจะมีการเก็บวัน/เวลาที่ผู้ใช้ตั้งหรือตอบกระทู้ เป็นต้น

Timestamp

- ภาษา PHP จะทำงานกับข้อมูล/เวลาในรูปแบบ Unix timestamp หรือสามารถเรียกสั้นๆได้ว่า Timestamp
- รูปแบบนี้จะเก็บวัน/เวลาเป็นเลขจำนวนเต็ม หรือ Integer ซึ่งค่านี้คือจำนวนวินาทีนับจากเวลาเที่ยงคืนของวันที่ 1 มกราคม ค.ศ.1970 ตามเวลามาตรฐานกรีนิช (Greenwich Mean Time – GMT)
- เช่นถ้า Timestamp มีค่าเท่ากับ 0 ก็จะหมายถึงวันและเวลาดังกล่าว ถ้ามีค่า 3600 จะหมายถึง เวลา 01.00 น. ของวันที่ 1 มกราคม ค.ศ.1970 และถ้ามีค่า -3600 จะหมายถึงเวลา 23.00 น. ของวันที่ 31 ธันวาคม ค.ศ.1969

การจัดรูปแบบวัน/เวลาเพื่อแสดงผล

- เมื่อทราบค่า Timestamp ที่เป็นตัวแทนของวัน/เวลาหนึ่งๆ ก็จะสามารถหาวัน เดือน ปี ชั่วโมง นาที และวินาที ของวัน/เวลานั้นได้ โดยอาศัยการคำนวณทางคณิตศาสตร์ หลังจากนั้นก็สามารถสร้างสตริงที่แสดงวัน/เวลาในรูปแบบตามที่ต้องการ
- ภาษา PHP มีฟังก์ชัน date ซึ่งสามารถหาและจัดรูปแบบวัน/เวลา โดยไม่ต้องทำการคำนวณเอง

ฟังก์ชันการจัดรูปแบบวัน/เวลาเพื่อแสดงผล - date

- ฟังก์ชัน date จะให้ผลลัพธ์เป็นค่าสตริงที่แสดงวัน/เวลาตามรูปแบบที่กำหนดโดยอาร์กิวเมนต์ตัวแรก โดยใช้ค่าวัน/เวลาจาก Timestamp ที่กำหนดโดยอาร์กิวเมนต์ตัวที่สอง หรือใช้ค่า Timestamp ของวัน/เวลาปัจจุบันหากไม่ระบุอาร์กิวเมนต์ตัวที่สอง
- รูปแบบมาตรฐานของฟังก์ชัน date

date(string format, [int timestamp])

- string format รูปแบบของวันที่และเวลาซึ่งจะแทนด้วยตัวอักษรภาษาอังกฤษหนึ่งตัว
- int timestamp เป็นตัวเลข integer ลำดับเวลาทุกหนึ่งวินาที นับตั้งแต่วันที่ 1 มกราคม 1970 เวลา 00:00:00 น. GMT

ฟังก์ชันการจัดรูปแบบวัน/เวลาเพื่อแสดงผล - date

- จากรูปแบบมาตรฐานอาร์กิวเมนต์ตัวแรกซึ่งเป็นสตริงกำหนดรูปแบบ (string format) นั้น ทำให้สามารถระบุตัวอักษรที่เป็น Format Character เพื่อแสดงส่วนต่างๆของวัน/เวลาที่กำหนดออกมาตรงตำแหน่งนั้นได้ เช่น ระบุตัวอักษร j เพื่อให้แสดงวันที่ หรือตัวอักษร F เพื่อให้แสดงชื่อเดือน เป็นต้น
- ตารางต่อไปจะแสดงถึงตัวอักษรที่ทำหน้าที่เป็น Format character ใน string format ของฟังก์ชัน date

ฟังก์ชันการจัดรูปแบบวัน/เวลาเพื่อแสดงผล - date

- ตารางแสดงตัวอักษรที่ทำหน้าที่เป็น Format character ใน string format ของฟังก์ชัน date

ตัวอักษร	คำอธิบาย	ค่าที่เป็นไปได้/ตัวอย่างค่า
d	เลขวันที่ โดยเติมศูนย์ข้างหน้าให้ครบ 2 หลัก	01 ถึง 31
D	ชื่อวันในสัปดาห์ เฉพาะ 3 ตัวอักษรแรก	Mon ถึง Sun
j	เลขวันที่ โดยไม่มีศูนย์ข้างหน้า	1 ถึง 31
l (L เล็ก)	ชื่อเต็มของวันในสัปดาห์	Sunday ถึง Saturday
N	เลขแสดงวันในสัปดาห์ตามมาตรฐาน ISO-8601	1 (วันจันทร์) ถึง 7 (วันอาทิตย์)
S	Ordinal suffix ของวันที่ (มักใช้ร่วมกับ j)	St, nd, rd หรือ th
w	เลขแสดงวันในสัปดาห์	0 (วันอาทิตย์) ถึง 6 (วันเสาร์)
z	หมายเลขวันของปี โดยเริ่มจาก 0 หมายถึงวันที่ 1 มกราคม	0 ถึง 365
W	หมายเลขสัปดาห์ของปีตามมาตรฐาน ISO-8601	01 ถึง 53
F	ชื่อเต็มของเดือน	January ถึง December

ฟังก์ชันการจัดรูปแบบวัน/เวลาเพื่อแสดงผล - date

- ตารางแสดงตัวอักษรที่ทำหน้าที่เป็น Format character ใน string format ของฟังก์ชัน date

ตัวอักษร	คำอธิบาย	ค่าที่เป็นไปได้/ตัวอย่างค่า
m	หมายเลขของเดือน โดยเติมศูนย์ข้างหน้าให้ครบ 2 หลัก	01 ถึง 12
M	ชื่อเดือน เฉพาะ 3 ตัวอักษรแรก	Jan ถึง Dec
n	หมายเลขของเดือน โดยไม่เติมศูนย์ข้างหน้า	1 ถึง 12
t	จำนวนวันของเดือน	28 ถึง 31
L	บ่งบอกปีอธิกสุรทิน	1 ใช่. 0 ไม่ใช่
Y	เลขปี ค.ศ. 4 หลัก	ตัวอย่าง: 1999 หรือ 2010
y	เลขปี ค.ศ. 2 หลัก	ตัวอย่าง: 99 หรือ 10
a	ตัวอักษรเล็กของ Ante Meridiem และ Post Meridiem	am หรือ pm
A	ตัวอักษรใหญ่ของ Ante Meridiem และ Post Meridiem	AM หรือ PM
g	เลขชั่วโมงแบบ 12 ชั่วโมง โดยไม่มีศูนย์นำหน้า	1 ถึง 12

ฟังก์ชันการจัดรูปแบบวัน/เวลาเพื่อแสดงผล - date

- ตารางแสดงตัวอักษรที่ทำหน้าที่เป็น Format character ใน string format ของฟังก์ชัน date

ตัวอักษร	คำอธิบาย	ค่าที่เป็นไปได้/ตัวอย่างค่า
Z	ความแตกต่างจากเวลามาตรฐานกรีนิชในหน่วยวินาที	ตั้งแต่ -43200 ถึง 43200
c	รูปแบบวัน/เวลาตามมาตรฐาน ISO-8601	ตัวอย่าง: 2006-10-26T13:18:56+07:00
r	รูปแบบวัน/เวลาตามมาตรฐาน RFC-2822	ตัวอย่าง: Thu, 26 Oct 2006 13:18:56 +0700
h	แสดงชั่วโมงในรูปแบบ 12 ชั่วโมง เป็นเลข 2 หลัก	01 ถึง 12
i	แสดงนาทีเป็นเลข 2 หลัก	00-59
s	แสดงวินาทีเป็นเลข 2 หลัก	00-59
A	แสดงช่วงเวลาเช้า หรือบ่าย เป็นตัวพิมพ์ใหญ่	AM หรือ PM

```
<?php  
echo date("Y/m/d") . "<br />";  
echo date("Y.m.d") . "<br />";  
echo date("Y-m-d") . "<br />";  
echo date("D d F Y เวลา h:i:s A");  
?>
```



```
2012/11/26  
2012.11.26  
2012-11-26  
Mon 26 November 2012 เวลา 03:15:46 PM
```

ฟังก์ชันการตรวจสอบวัน - checkdate

- ในการบันทึกข้อมูลไม่ว่าจะเป็นแบบสอบถาม การสมัครสมาชิก การลงทะเบียน มักมีข้อมูลวันที่เข้าไปเกี่ยวข้องด้วยเสมอ จึงเป็นหน้าที่ของผู้เขียนโปรแกรมที่จะต้องตรวจสอบความถูกต้องของข้อมูลวันที่ก่อนทำการบันทึกลงฐานข้อมูล (รูปแบบวันที่ที่ MySQL รับได้คือ yyyy-mm-dd, yy-mm-dd)
- PHP มีฟังก์ชัน checkdate ซึ่งทำหน้าที่ตรวจสอบความถูกต้องของข้อมูลประเภทวันที่ โดยมีรูปแบบมาตรฐานดังนี้

checkdate(int month, int day, int year)

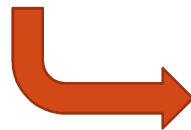
- int month ตัวเลขระบุเดือน
- int day ตัวเลขระบุวัน
- int year ตัวเลขระบุปี

ฟังก์ชันการตรวจสอบวัน - checkdate

- ตัวอย่างการใช้ฟังก์ชัน checkdate

```
<?php
echo "ตรวจสอบวันที่ 29 กุมภาพันธ์ ค.ศ.2005". "<br>";
if(checkdate(2, 29, 2005))
    echo "วันเดือนปีนี้ใช้ได้ (มีอยู่จริง)". "<br>";
else
    echo "วันเดือนปีนี้ใช้ไม่ได้ (ไม่มีอยู่จริง)". "<br>";
echo "<br>";

echo "ตรวจสอบวันที่ 18 ตุลาคม ค.ศ.2004". "<br>";
if(checkdate(10, 18, 2004))
    echo "วันเดือนปีนี้ใช้ได้ (มีอยู่จริง)". "<br>";
else
    echo "วันเดือนปีนี้ใช้ไม่ได้ (ไม่มีอยู่จริง)". "<br>";
echo "<br>";
?>
```



ตรวจสอบวันที่ 29 กุมภาพันธ์ ค.ศ.2005
วันเดือนปีนี้ใช้ไม่ได้ (ไม่มีอยู่จริง)

ตรวจสอบวันที่ 18 ตุลาคม ค.ศ.2004
วันเดือนปีนี้ใช้ได้ (มีอยู่จริง)