

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยขอนแก่น

PHP Crash Course

วัตถุประสงค์

- สามารถสร้างโปรแกรมในภาษา PHP และสั่งให้ทำงานได้
- รู้จักคำสั่งพื้นฐานที่สำคัญในภาษา PHP

1. ภาษา PHP คืออะไร

PHP คือภาษา script อย่างหนึ่งที่เป็น server-side script ซึ่งจะทำงานในฝั่ง server แล้วส่งการแสดงผลมายัง browser ของตัว Client และนอกจากนี้ มันยังเป็น script ที่ embed บน HTML อีกด้วย

PHP เป็นภาษาจำพวก script language คำสั่งต่าง ๆ จะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ตัวอย่างของภาษาสคริปต์เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่น ๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาโดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded. Scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพมากขึ้น

PHP ได้รับการเผยแพร่เป็นครั้งแรกในปี ค.ศ. 1994 โดยคุณ Rasmus Lerdorf ต่อมาได้มีนักโปรแกรมเมอร์เข้ามาช่วยในการพัฒนาต่อมาตามลำดับ เป็นเวอร์ชันต่าง ๆ จนกระทั่งถึงเวอร์ชันล่าสุดซึ่งเป็นเวอร์ชัน 5 นักพัฒนาสำคัญของเวอร์ชัน 4 และ เวอร์ชัน 5 คือคุณ Zeev Suraski และคุณ Andi Gutmans ในขณะนี้ มีเว็บไซต์เวอร์ประมาณ 16 ล้านโดเมน (domains) ที่ใช้ PHP เราสามารถตรวจสอบจำนวนของ domains ที่ใช้ PHP ได้ที่

<http://www.php.net/usage.php>

ในตอนแรก PHP ย่อมาจาก Personal Home Page แต่ต่อมาได้เปลี่ยนชื่อให้ตรงตามกฎเกณฑ์ของ GNU ชื่อในปัจจุบันของ PHP นั้นย่อมาจาก Hypertext Preprocessor รายละเอียดต่าง ๆ ของ PHP เราสามารถเข้าไปค้นหาเพิ่มเติมได้ที่เว็บไซต์ของ PHP ซึ่งคือ

<http://www.php.net>

## 2. ทำไมภาษา PHP น่าสนใจและน่าใช้

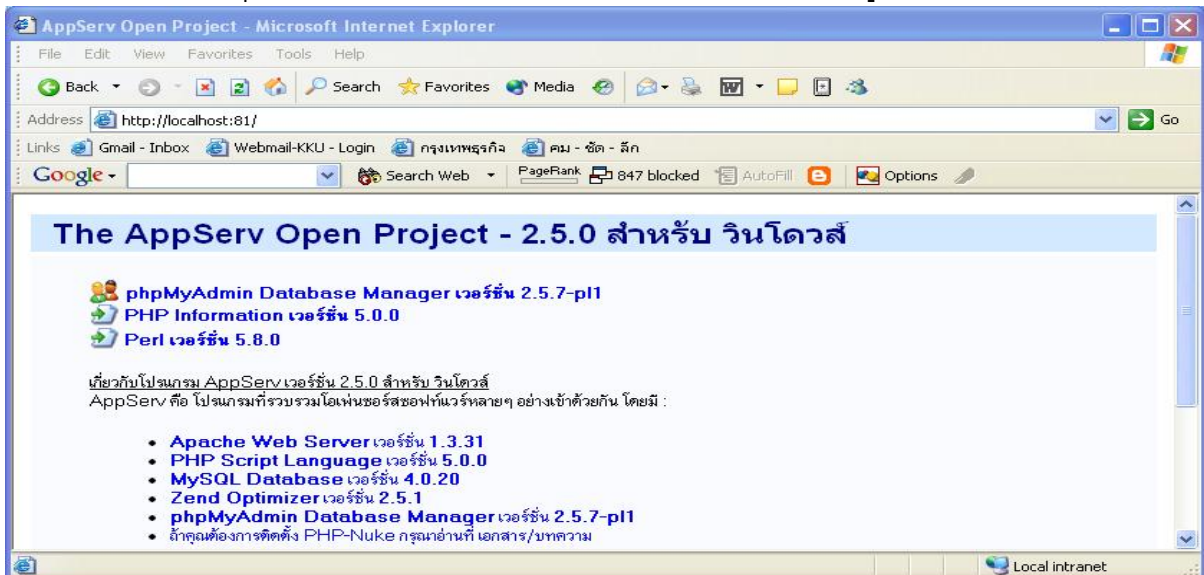
ภาษาอื่นที่ทำหน้าที่คล้าย ๆ กับภาษา PHP คือ Perl, Microsoft Active Server Pages (ASP), Java Server Page (JSP), และ Allaire ColdFusion

ถ้าเปรียบเทียบภาษา PHP กับ ภาษาอื่น ๆ เหล่านี้เราจะพบว่าภาษา PHP มีข้อได้เปรียบหลายอย่างดังต่อไปนี้

- มีสมรรถนะสูง: สามารถรองรับการใช้หลายล้าน hits ในแต่ละวัน
- สามารถติดต่อกับหลายประเภทของฐานข้อมูลอย่างเช่น MySQL, PostgreSQL, mSQL, Oracle, Informix, Sybase และสามารถใช้ Open Database Connectivity Standard (ODBC) เพื่อติดต่อกับผลิตภัณฑ์ฐานข้อมูลของ Microsoft
- ไม่ต้องเสียค่าใช้จ่ายในการใช้: เราสามารถดาวน์โหลด PHP ได้จาก <http://www.php.net> โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ
- เรียนรู้และใช้ง่าย โดยเฉพาะถ้าเรารู้ภาษา C++, Perl, และ Java อยู่แล้ว
- สามารถใช้ PHP ได้บนหลายระบบปฏิบัติการโดยที่ไม่ต้องเปลี่ยนโปรแกรม

## 3. การติดตั้ง PHP and MySQL

เราจะใช้ AppServ ซึ่งจะ Install PHP and MySQL ให้เรา เราสามารถดาวน์โหลด AppServ ได้จาก <http://gear.kku.ac.th/~krunapon/downloads/appserv> หากการติดตั้งสำเร็จเรียบร้อยดี คุณควรจะเห็นเว็บเพจที่ <http://localhost/> ปรากฏดังในรูปที่ 1



รูปที่ 1 The AppServ Open Project

ในรูปภาพนี้ผู้ติดตั้ง appserv ตั้ง port ที่ 81 ดังนั้น URL ในรูปที่ 1 จึงเป็น <http://localhost:81/>

#### 4. ความรู้เบื้องต้นเกี่ยวกับภาษา PHP

##### 4.1 การแทรกคำสั่งภาษา PHP ในเอกสาร HTML

เราสามารถแทรกคำสั่งภาษา PHP ได้โดยการใช้ PHP tags ซึ่งมีไว้เพื่อเป็นการบ่งบอกให้รู้ว่าส่วนที่เป็นคำสั่งของภาษา PHP

ที่อยู่ในเอกสาร HTML การใช้ PHP tags นั้นสามารถทำได้หลายรูปแบบดังต่อไปนี้

- `<?php ... ?>` (XML style)
- `<? ... ?>` (short style)
- `<script language='php'>...</script>` (Script style)
- `<% ... %>` (ASP style)

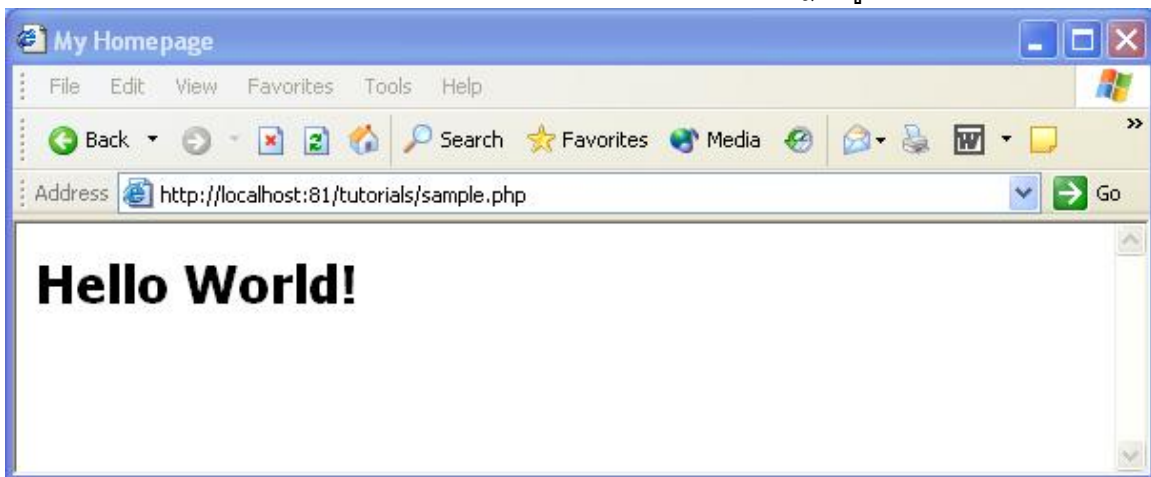
แบบที่ควรใช้คือ XML style เนื่องจากสามารถรันได้กับทุก Server อีกทั้งสอดคล้องกับไวยากรณ์ของภาษา XML

เราสามารถวางคำสั่งในภาษา PHP ไว้ภายในเอกสาร HTML ตามที่ต้องการได้ โดยที่ PHP tags อาจจะวางอยู่สลับกับ HTML tags ดังตัวอย่างเช่น

##### **Example1: sample.php**

```
<html>
  <head><title>My Homepage</title></head>
  <body><h1><?php echo "Hello World!"; ?></h1></body>
</html>
```

ผลลัพธ์ที่ได้จากการใช้โปรแกรม sample.php ปรากฏในรูปที่ 2



รูปที่ 2 sample.php

## 4.2 รูปแบบภาษา PHP และคำสั่ง

### 4.2.1 Comments

Comments ในภาษา PHP เอาจรูปแบบมาจาก comments ในภาษา C++ และภาษา XML

Comments มีไว้เพื่อเป็น บันทึกช่วยให้ผู้เขียนหรือผู้อ่านโปรแกรมเข้าใจโปรแกรม ตัวอย่างของ comments มีดังนี้

```
<!-- file: hello.php (comments in xml style) -->

// this style of comment is suitable for a short comment that is not over one line

/*
 * to write comments more than one line, you may want to use this style
 */
```

### 4.2.2 Variables (ตัวแปร)

Variables หรือ ตัวแปร คือ สัญลักษณ์ที่ใช้แทนข้อมูล Literals คือข้อมูลซึ่งสามารถเป็นค่า ของตัวแปร \$salary เป็นตัวแปร 20,000 เป็น Literal วิธีการกำหนดตัวแปรในภาษา PHP จะต้องขึ้นต้นด้วยเครื่องหมาย \$ แล้วตามด้วยชื่อของตัวแปร

ตัวอย่างการใช้ตัวแปร

#### **Example 2: variable.php**

```
<?php
    $salary = 20000;
    echo "Your salary is $salary Bahts.";
?>
```

ผลลัพธ์ที่ได้จากการใช้โปรแกรม variable.php ปรากฏในรูปที่ 3



รูปที่ 3 variable.php

ถ้าหากว่าคุณไปที่ “View” และ คลิกที่ “Source” คุณจะพบว่าเนื้อหาของ file “variable.php” เป็น “Your salary is 20000 Bahts” ไม่มีคำสั่งของภาษา PHP ซึ่งหมายความว่า Server ได้รับและแปลคำสั่งของคุณเรียบร้อยแล้ว ไม่ว่า Browsers ใดก็จะเห็นเนื้อหาของ file นี้เหมือนกันหมด

ตัวแปรในภาษา PHP นั้นมีอยู่หลายชนิดดังต่อไปนี้

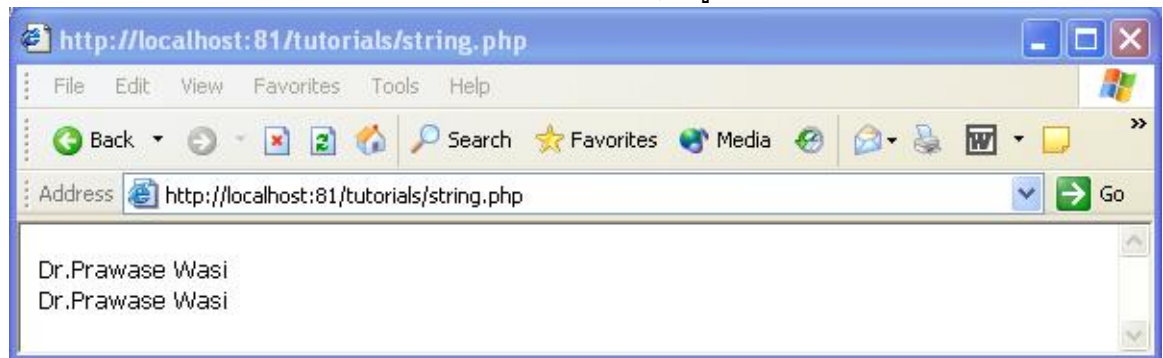
1. String
2. Integer, Long Integer
3. Float, Real, Double
4. Array
5. Object

ตัวอย่างการใช้ตัวแปรชนิด String

### **Example 3: string.php**

```
<?php
    $prefix="Dr.";
    $firstname="Prawase";
    $surname="Wasi";
    echo $prefix.$firstname." ".$surname."<br/>";
    echo "$prefix$firstname $surname<br/>";
?>
```

ผลลัพธ์ที่ได้จากการใช้โปรแกรม string.php ปรากฏในรูปที่ 4



รูปที่ 4 string.php

จากตัวอย่างข้างบนเราเรียนรู้ว่าเราสามารถเอาค่าของตัวแปรชนิด String มาติดต่อกัน โดยใช้ “.” Operator. นอกจากนี้เราพบว่าค่าของตัวแปรจะถูกประเมินถ้าหากว่าเราเอาตัวแปรใส่

ในเครื่องหมายคำพูด (“...”) ถ้าหากว่าเราต้องการพิมพ์ String “\$firstname” ออกมาโดยที่ไม่ได้ต้องการให้ PHP เข้าใจว่านั่นคือตัวแปรและพิมพ์ค่าของตัวแปรออกมา เราต้องใช้เครื่องหมาย ‘...’

PHP อนุญาตให้เราใช้ References. Reference operator คือ & ซึ่งจะใช้กับเครื่องหมาย = เพื่อจะ copy address ของ variables จะขอยกตัวอย่างต่อไปนี้เพื่อให้เข้าใจมากขึ้น

```
$a = 5;
$b = $a;
$a = 7; // $b will still be 5
$a = 5;
$b = &$a;
$a = 7; // both $a and $b are now both 7
```

#### 4.2.3 Operators

Operators ที่เราสามารถใช้ได้ภาษา PHP มีดังนี้

- a) Arithmetic operators: +, -, \*, /, %
- b) Combined assignment operators: +=, -=, \*=, /=, %=, .=
- c) Pre- and post- increment and decrement: ++, ++a, --, --a;
- d) Comparison operators: == (equals), === (identical), != (not equal), <> (not equal), <, >, <=, >=

Operator === จะ return ค่า true ก็ต่อเมื่อ ตัวแปรสองตัวแปรมีค่าเท่ากันและเป็นตัวแปรที่อยู่ในชนิดเดียวกัน ตัวอย่างต่อไปนี้จะทำให้เห็นข้อแตกต่างระหว่าง operator “==” และ operator “===”

#### Example 4: equal\_identical.php

```
<?php
    $a = 5.0;
    $b = 5;
    if ($a == $b)
        echo "$a == $b<br/>";
    if ($a === $b)
        echo "$a === $b<br/>";
    else
        echo "$a not === $b<br/>";
?>
```

ซึ่งจะได้ผลลัพธ์ดังปรากฏในรูปที่ 5



รูปที่ 5 equal\_identical.php

e) Logical operators: !, &&, ||, and, or

and และ or operators ให้ผลเช่นเดียวกับ && และ || แต่ && และ || จะถูกประเมินผลก่อน

f) Bitwise operators: & (AND) , | (OR) , ~ (NOT) , ^ (XOR), << (left shift) , >> (right shift)

#### Example 5: bit\_operations.php

```
<?php
    $a = 9; // 10012
    $b = 3; // 00112
    echo '$a.' is $a and '.$b.' is $b<br/>';
    $c = $a | $b; // 10012 | 00112 = 10112 → 11
    echo '$a | $b =' . $c<br/>';
```

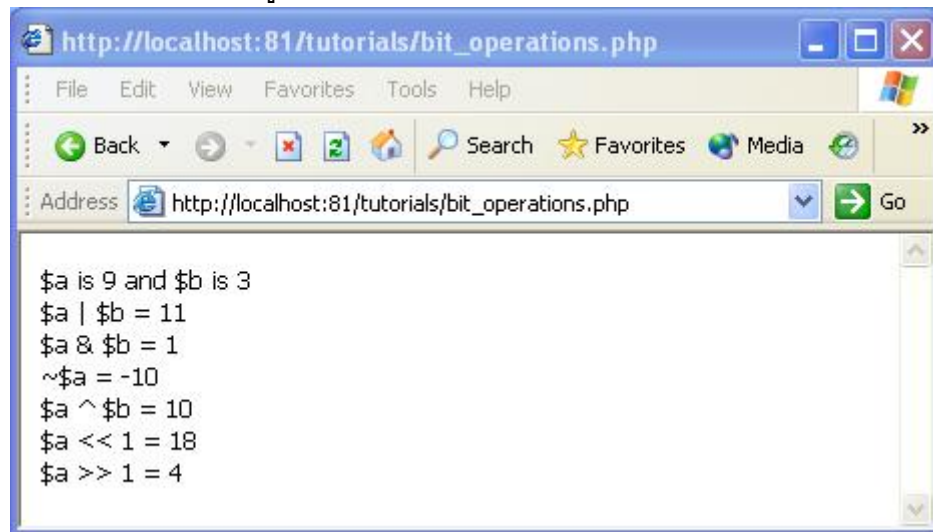
```

$d = $a & $b; // 10012 | 00112 = 00012 → 1
echo '$a & $b = ' . $d <br/>";
$e = ~$a; // ~(010012) = 10110 → -10
echo '~$a = ' . $e <br/>";
$f = $a ^ $b; // 10012 | 00112 = 10102 → 10
echo '$a ^ $b = ' . $f <br/>";
$g = $a << 1; // 10012 << 1 = 100102 → 18
echo '$a << 1 = ' . $g <br/>";
$h = $a >> 1; // 10012 >> 1 = 01002 → 4
echo '$a >> 1 = ' . $h <br/>";

```

?>

ผลลัพธ์มีดังปรากฏในรูปที่ 6



รูปที่ 6 bit\_operations.php

#### 4.2.4 คำสั่ง if/else

คำสั่ง if/else เป็นคำสั่งที่เลือกการทำงานตามเงื่อนไข รูปแบบของคำสั่งมีดังต่อไปนี้

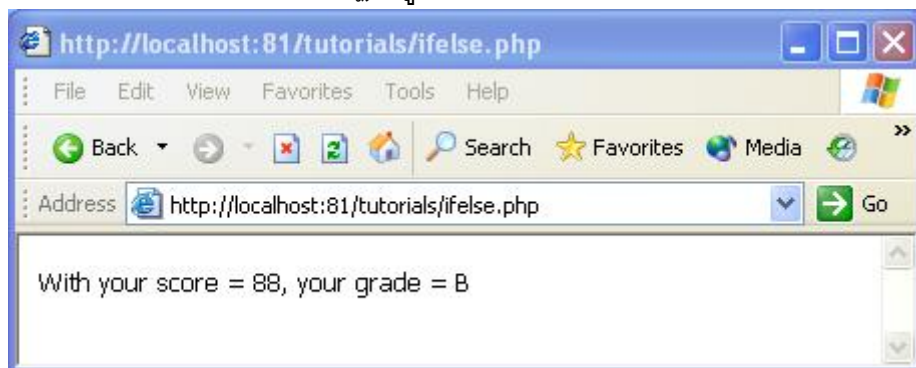
- if (เงื่อนไข) { คำสั่ง }
- if (เงื่อนไข) { คำสั่ง } else { คำสั่ง }
- if (เงื่อนไข) { คำสั่ง } elseif { คำสั่ง } else { คำสั่ง }

ตัวอย่างของการใช้คำสั่ง if/else มีดังต่อไปนี้

### **Example 6: ifelse.php**

```
<?php
    $score = 88;
    $grade = 'F';
    if ($score > 60 && $score <= 70)
    {
        $grade = 'D';
    }
    elseif ($score > 70 && $score <= 80)
    {
        $grade = 'C';
    }
    elseif ($score > 80 && $score <= 90)
    {
        $grade = 'B';
    }
    else
    {
        $grade = 'A';
    }
    echo "With your score = $score, your grade = $grade<br/>";
?>
```

ผลลัพธ์ของโปรแกรมดังปรากฏในรูปที่ 7



รูปที่ 7 ifelse.php

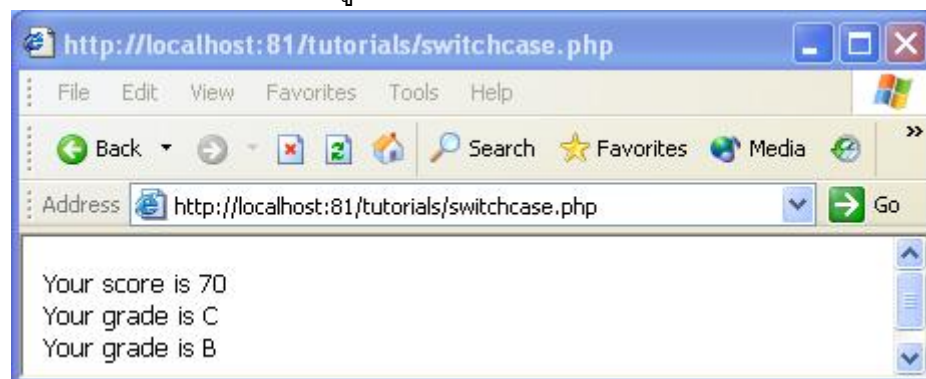
#### 4.2.5 คำสั่ง switch/case

คำสั่ง switch/case เป็นคำสั่งที่เลือกการทำงานตามเงื่อนไขเช่นเดียวกับคำสั่ง if/else แต่ว่า นิพจน์ของ if เป็น expression อย่างเช่น (a == 2) หรือ (a\*b > 5) ได้ ในขณะที่ นิพจน์ของ case ต้องเป็นค่าคงที่ที่เป็นจำนวนเต็มเท่านั้น (อย่างเช่น 2, 'a', 2.7 แต่ไม่ใช่ 2.5) และเช็คเฉพาะว่าตัวแปรมีค่าเท่ากับจำนวนเต็มที่ให้มาหรือไม่ (เช่นไม่สามารถเช็คได้ว่า a > 2 หรือไม่) อีกประการหนึ่ง คำสั่งใน switch ถ้าหากว่าถูกกระทำแล้วจะหยุดก็ต่อเมื่อเจอ break

##### **Example 7: switchcase.php**

```
<?php
    $score = 70;
    $grade = 'F';
    echo "Your score is $score<br/>";
    switch ($score)
    {
        case 60:
            echo "Your grade is D<br/>";
        case 70:
            echo "Your grade is C<br/>";
        case 80:
            echo "Your grade is B<br/>";
            break;
        case 90:
            echo "Your grade is A<br/>";
    }
?>
```

ผลลัพธ์ที่ได้ของโปรแกรมในรูปแบบที่ 8



รูปที่ 8 switchcase.php

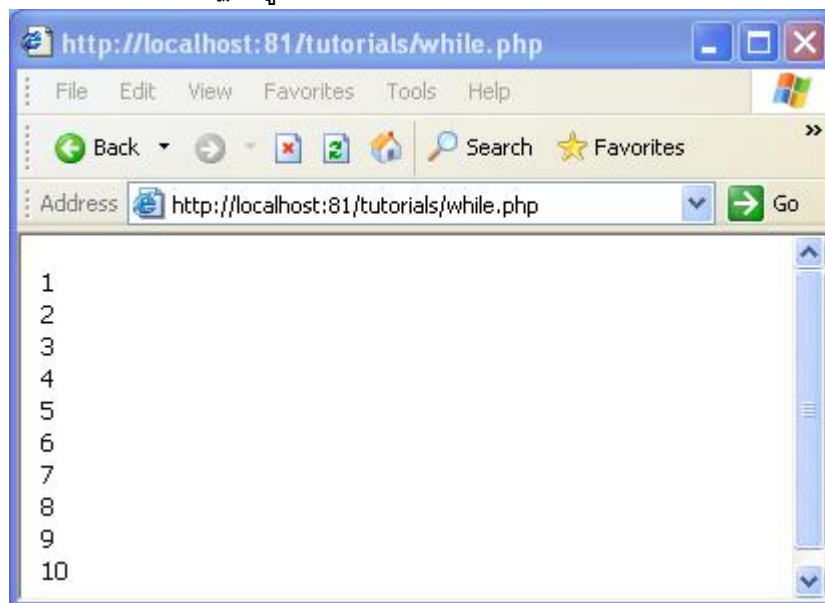
#### 4.2.6 คำสั่ง while

คำสั่ง while เป็นคำสั่งวนลูปตามเงื่อนไข รูปแบบของคำสั่งคือ  
while (เงื่อนไข) { คำสั่ง }

##### **Example 8: while.php**

```
<?php
    $i = 1;
    while ($i <= 10)
    {
        echo "$i<br/>";
        $i++;
    }
?>
```

ผลลัพธ์ที่ได้ปรากฏในรูปที่ 9



รูปที่ 9 while.php

#### 4.2.7 คำสั่ง do/while

คำสั่ง do/while เป็นคำสั่งวนลูป ทำงานตามเงื่อนไข รูปแบบของคำสั่งคือ  
do { คำสั่ง } while (เงื่อนไข)

### **Example 9: dowhile.php**

```
<?php
    $i = 1;
    do {
        echo "$i<br/>";
        $i++;
    } while ($i <= 10)
?>
```

ผลลัพธ์ที่ได้จะเป็นเช่นเดียวกับที่ได้จากการรันโปรแกรม while.php

### **4.2.8 คำสั่ง for**

เป็นคำสั่งที่เลือกการทำงานตามเงื่อนไข รูปแบบของคำสั่งคือ  
for (ค่าเริ่มต้น; ค่าสิ้นสุด; เงื่อนไข) { คำสั่ง }

### **Example 10: for.php**

```
<?php
    for ($i = 1; $i <= 10; $i++)
    {
        echo "$i<br/>";
    }
?>
```

ผลลัพธ์ที่ได้จะเป็นเช่นเดียวกับที่ได้จากการรันโปรแกรม while.php และ  
dowhile.php

### **4.2.9 การกำหนดตัวแปรที่มีค่าคงที่**

การกำหนดตัวแปรที่มีค่าคงที่ใน PHP ทำได้โดยการใช้คำสั่ง define  
ดังเช่น

```
define('MANGOPRICE', 20);
```

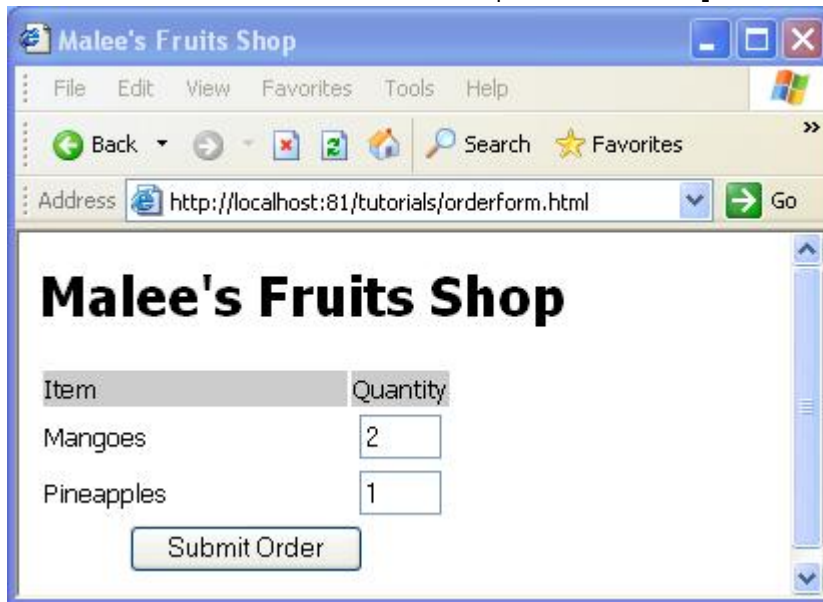
ตัวแปรเหล่านี้ควรจะเป็นตัวพิมพ์ใหญ่หมดและเวลาอ้างอิงถึงไม่ต้องมี \$  
นำหน้า

## การทดลอง

### 1. เขียนโปรแกรม orderform.html ดังต่อไปนี้

```
<html>
  <head>
    <title>Malee's Fruits Shop</title>
  </head>
  <body>
    <h1>Malee's Fruits Shop</h1>
    <form action="processorder.php" method="post">
      <table border="0">
        <tr bgcolor="#cccccc">
          <td width="150">Item</td>
          <td width="15">Quantity</td>
        </tr>
        <tr>
          <td>Mangoes</td>
          <td align="center">
            <input type="text"
              name="mangoqty" size="3"
              maxlength="3">
          </td>
        </tr>
        <tr>
          <td>Pineapples</td>
          <td align="center">
            <input type="text"
              name="pineappleqty" size="3"
              maxlength="3">
          </td>
        </tr>
        <tr>
          <td colspan="2" align="center">
            <input type="submit" value="Submit
              Order"></td>
        </tr>
      </table></form></body></html>
```

จากการเขียนโปรแกรม HTML ข้างบนนี้คุณควรจะได้ดังในรูปที่ 10



รูปที่ 10 orderform.html

- เขียน PHP code ใน file “processorder.php”
  - โดยที่คุณสามารถจะรู้จำนวน mangoes จาก variable `$mangoqty` และรู้จำนวน pineapples จาก variable `$pineappleqty`
  - ใช้การกำหนดตัวแปรทั้งที่มีค่าไม่คงที่และค่าคงที่เพื่อจะได้ผลดังข้างล่างนี้ โดยที่กำหนดว่ามะม่วงลูกละ 10 บาท และสับปะรดลูกละ 35 บาท คุณจะต้องกำหนดราคาของผลไม้ในรูปของตัวแปรที่มีค่าคงที่
  - เขียน PHP codes เพื่อบอกว่าลูกค้ายเป็นลูกค้าชั้นเยี่ยม (excellent) ถ้าหากว่าเขาซื้อผลไม้มากกว่า 50 Bahts ลูกค้ายเป็นลูกค้าชั้นดีมาก (very good) ถ้าหากเขาซื้อผลไม้มากกว่า 40 Bahts. ลูกค้ายเป็นลูกค้าชั้นดี (good) ถ้าหากเขาซื้อผลไม้มากกว่า 30 Bahts แต่ไม่ว่าลูกค้ายจะซื้อสินค้าจากคุณจำนวนเท่าใด คุณก็เขียนขอบคุณเขา
  - พิมพ์ดาว 5 ดวงสำหรับลูกค้ายที่เป็นลูกค้าชั้นเยี่ยม (excellent) ดาว 4 ดวงสำหรับลูกค้ายที่เป็นลูกค้าชั้นดีมาก (very good) ดาว 3 ดวงสำหรับลูกค้ายที่เป็นลูกค้าชั้นดี (good) โดยที่การพิมพ์ดาวจะต้องใช้ “for” หรือ “while” หรือ “do ... while” loops

ถ้าหากว่าซื้อมะม่วง 2 ผลและแอปเปิ้ล 1 ผล คุณจะได้ผลลัพธ์ดังในรูปที่ 11



รูปที่ 11: 2 mangoes and 1 pineapples

ถ้าหากว่าซื้อมะม่วง 1 ผลและแอปเปิ้ล 0 ผล คุณจะได้อผลลัพธ์ดังในรูปที่ 12



รูปที่ 12: 1 mangoes and 0 pineapples

ถ้าหากว่าซื้อมะม่วง 0 ผลและแอปเปิ้ล 0 ผล คุณจะแสดงผลดังในรูปที่ 13



รูปที่ 13: 0 mangoes and 0 pineapples