บทที่ 4
โครงสร้างควบคุม

โครงสร้างควบคุม (Control Structures) คือ โครงสร้างที่ใช้สำหรับควบคุมการไหลของสคริปต์คำสั่ง หรือกำหนดทางเลือกให้กระทำหรือไม่กระทำ เช่น การตรวจสอบเงื่อนไข การทำคำสั่งซ้ำๆ ตามเงื่อนไขกำหนด เป็นต้น ลักษณะการทำงานดังกล่าวนี้ถือเป็นพื้นฐานที่สำคัญที่ผู้เขียนโปรแกรมจำเป็นต้องเรียนรู้ ดังนั้นเนื้อหาในบทนี้จะได้เรียนรู้เกี่ยวกับโครงสร้างการควบคุมดังกล่าวที่สามารถใช้ได้ในภาษา PHP

โครงสร้างควบคุม สามารถแบ่งออกเป็น 3 กลุ่ม ประกอบด้วย 1) โครงสร้างเงื่อนไข (Conditional Structures) 2) โครงสร้างเงื่อนไขทำซ้ำ (Loop Structures) และ 3) คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ มีรายละเอียด ดังนี้

หมายเหตุ

รูปแบบโครงสร้างควบคุมของ if, while, for, foreach และ switch สามารถเปลี่ยนจากเครื่องหมายปีกกาเปิด (bracket) แทนด้วยเครื่องหมาย colon (:) และแทนปีกกาปิดด้วย endif;, endwhile;, endfor;, endforeach; และ endswitch; ตามโครงสร้างควบคุมที่ใช้งาน

4.1 โครงสร้างเงื่อนไข

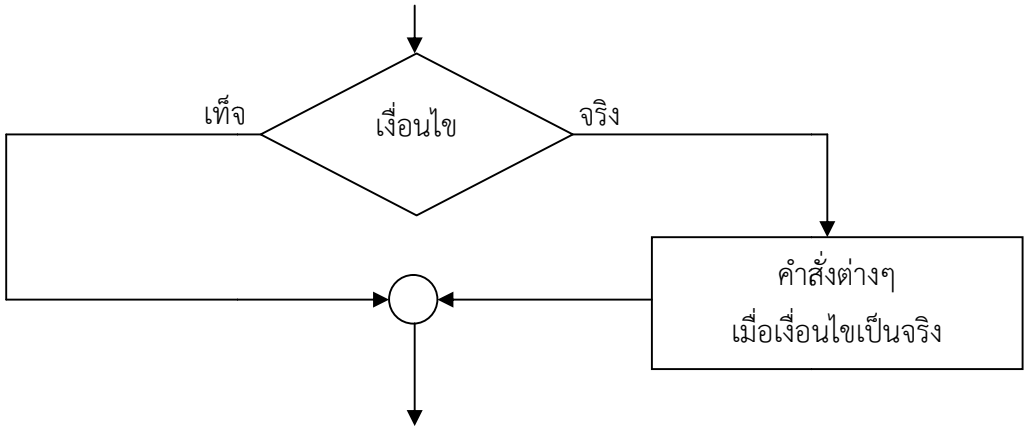
โครงสร้างเงื่อนไขเป็นหนึ่งในโครงสร้างควบคุมขั้นพื้นฐาน เพื่อกำหนดทางเลือกต่างๆ เพื่อให้โปรแกรมที่พัฒนาสามารถตอบสนองได้หลากหลายตามค่าที่รับเข้ามาหรือเงื่อนไขกำหนด โดยใช้รูปแบบการเปรียบเทียบทางตรรกะ โครงสร้างพื้นฐานของกลุ่มนี้ ประกอบด้วย 1) if 2) if...else 3) if ... elseif และ 4) switch มีรายละเอียด ดังนี้

4.1.1 โครงสร้างเงื่อนไข if (The if Statement)

โครงสร้างเงื่อนไข if ใช้ในการเปรียบเทียบเงื่อนไขที่ใช้สำหรับตัดสินใจ ถ้าเป็นจริงจะทำตามคำสั่งต่างๆ ที่กำหนดไว้ภายใต้เงื่อนไข ถ้าเป็นเท็จก็จะทำคำสั่งอื่นๆ หลังจากประโยคปิดโครงสร้างเงื่อนไข if โดยโครงสร้างเงื่อนไข if มีรูปแบบ ดังนี้

รูปแบบ

```
if (เงื่อนไข) {
        คำสั่งเมื่อเงื่อนไขเป็นจริง
}
```



ภาพที่ 4.1 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if

ตัวอย่างที่ 4.1 การใช้โครงสร้างเงื่อนไข if ในการเปรียบเทียบ

```
1   <?php
2       $a = 10;
3       $b = 5;
4       if ($a > $b) {
5           echo "a is bigger than b";
6       }
7   ?>
```

จากตัวอย่างที่ 4.1 การใช้โครงสร้างเงื่อนไข if ในการเปรียบเทียบ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $a มีค่าเท่ากับ 10

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร $b มีค่าเท่ากับ 5

บรรทัดที่ 4 โครงสร้างเงื่อนไข if เปรียบเทียบค่าตัวแปร $a มากกว่า ตัวแปร $b จริงหรือไม่ ถ้าเป็นจริงเข้าสู่การทำงานภายในโครงสร้างเงื่อนไข if คือ ทำคำสั่งในบรรทัดที่ 5 ถ้าเป็นเท็จออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 7

บรรทัดที่ 5 แสดงข้อความ a is bigger then b

บรรทัดที่ 6 ประโยคปิดโครงสร้างเงื่อนไข if

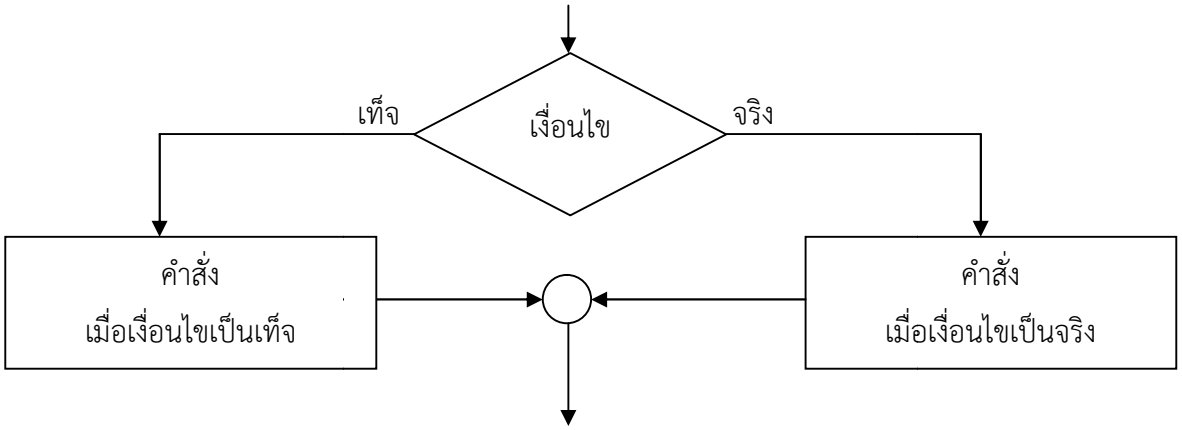
บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

4.1.2 โครงสร้างเงื่อนไข if ... else (The if ... else Statement)

โครงสร้างเงื่อนไข if...else เป็นโครงสร้างเงื่อนไขที่ใช้เปรียบเทียบเช่นเดียวกับ โครงสร้างเงื่อนไข if แต่มากกว่าตรงที่ โครงสร้างเงื่อนไข if .. else ถ้าเปรียบเทียบแล้วเป็นจริงก็จะให้ทำตามเงื่อนไขภายในโครงสร้างเงื่อนไข if ถ้าเป็นเท็จก็จะทำคำสั่งภายในโครงสร้างเงื่อนไข else มีรูปแบบ ดังนี้

รูปแบบ

```
if (เงื่อนไข) {
        คำสั่งเมื่อเงื่อนไขเป็นจริง
} else {
        คำสั่งเมื่อเงื่อนไขเป็นเท็จ
}
```



ภาพที่ 4.2 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if ... else

จากโครงสร้างเงื่อนไข if ก่อนหน้านี้จะเห็นว่า เมื่อเปรียบเทียบค่าตัวแปร $a มากกว่าค่าตัวแปร $b จริงก็จะทำตามคำสั่งที่อยู่ภายในโครงสร้างเงื่อนไข if แต่หากค่าตัวแปร $a น้อยกว่า หรือน้อยกว่าหรือเท่ากับ โปรแกรมจะไม่แสดงข้อความใดๆ หากประยุกต์ใช้โครงสร้างเงื่อนไข if ... else เพื่อแสดงข้อความเพิ่มเติม เช่น กรณีที่เป็นเท็จจากเงื่อนไข ในโครงสร้างเงื่อนไข if ให้แสดงข้อความอื่นๆ ตัวอย่างดังนี้

ตัวอย่างที่ 4.2 การใช้โครงสร้างเงื่อนไข if...else ในการเปรียบเทียบ

```
1   <?php
2           $a = 10;
3           $b = 5;
4           if ($a > $b) {
5                   echo "a is bigger than b";
6           } else {
7                   echo "a is less than or equal b";
8           }
9   ?>
```

จากตัวอย่างที่ 4.2 การใช้โครงสร้างเงื่อนไข if....else ในการเปรียบเทียบ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $a มีค่าเท่ากับ 10

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร $b มีค่าเท่ากับ 5

บรรทัดที่ 4 โครงสร้างเงื่อนไข if เปรียบเทียบค่าตัวแปร $a มากกว่า ตัวแปร $b จริงหรือไม่ ถ้าเป็นจริงเข้าสู่การทำงานภายในโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 5 ถ้าเป็นเท็จจะออกจากประโยคปิดโครงสร้างเงื่อนไข if ในบรรทัดที่ 6 และเข้าสู่โครงสร้างเงื่อนไข else

บรรทัดที่ 5 แสดงข้อความ a is bigger then b

บรรทัดที่ 6 ประโยคปิดโครงสร้างเงื่อนไข if และเริ่มต้นโครงสร้างเงื่อนไข else

บรรทัดที่ 7 แสดงข้อความ a is less than or equal b

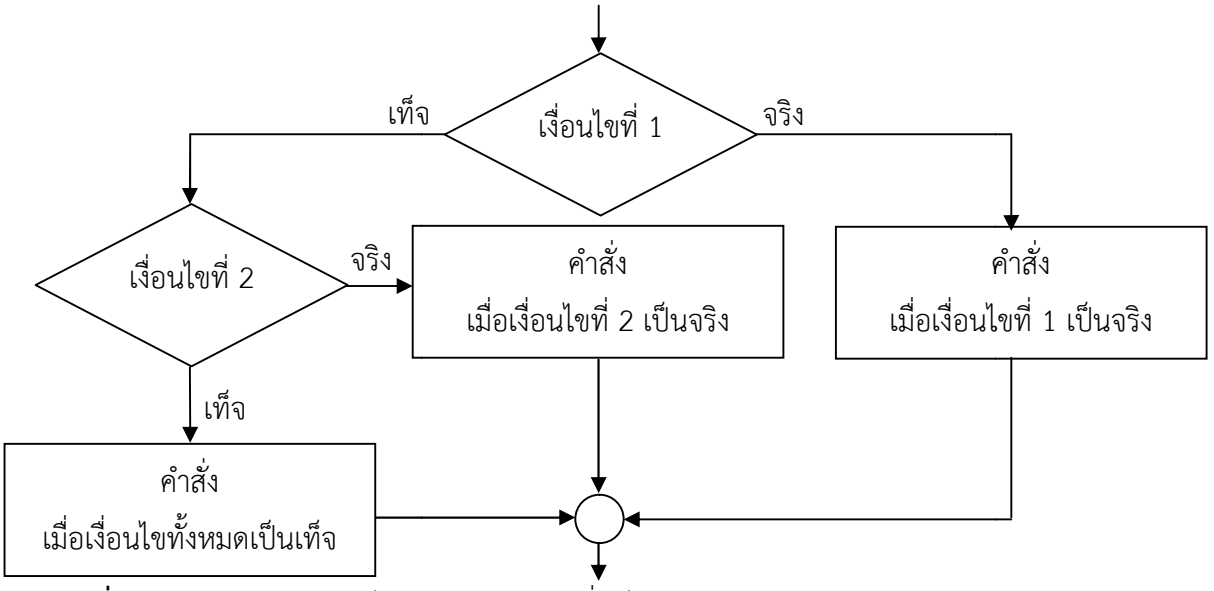
บรรทัดที่ 8 ประโยคปิดโครงสร้างเงื่อนไข else

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

4.1.3 โครงสร้างเงื่อนไข if ... elseif (The if ... elseif Statement)

โครงสร้างเงื่อนไข if ... elseif หรือ if ... else if เป็นโครงสร้างเงื่อนไขที่ใช้เปรียบเทียบเงื่อนไขมากกว่า 1 เงื่อนไข เป็นการรวมกันของโครงสร้างเงื่อนไข if และ else ซ้อนกันเรียงตามลำดับ มีรูปแบบ ดังนี้

```
if (เงื่อนไขที่ 1) {
        คำสั่งเมื่อเงื่อนไขเป็นจริง
} elseif (เงื่อนไขที่ 2) {          // หรือเว้นวรรคเป็น else if (เงื่อนไขที่ 2)
        คำสั่งเมื่อเงื่อนไขที่ 2 เป็นจริง
} else {
        คำสั่งเมื่อเงื่อนไขที่ 2 เป็นเท็จ
}
```



ภาพที่ 4.3 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if ... elseif

ตัวอย่างที่ 4.3 การใช้โครงสร้างเงื่อนไข if...elseif ในการเปรียบเทียบหลายเงื่อนไข

```
1   <?php
2       $score = 78;
3       if ($score>=80) {
4           $grade="A";
5       } else if ($score>=70) {
6           $grade="B";
7       } else if ($score>=60) {
8           $grade="C";
9       } else if ($score>=50) {
10          $grade="D";
11      } else {
12          $grade="E";
13      }
14      printf ("Score is %d: Grade evaluation is %s.", $score, $grade);
15  ?>
```

ผลลัพธ์

```
Score is 78: Grade evaluation is B.
```

จากตัวอย่างที่ 4.3 การใช้โครงสร้างเงื่อนไข if...elseif ในการเปรียบเทียบหลายเงื่อนไข อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $score มีค่าเท่ากับ 78

บรรทัดที่ 3 โครงสร้างเงื่อนไข if (เงื่อนไขที่ 1) เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 80 จริงหรือไม่ ถ้าเป็นจริงทำคำสั่งในบรรทัดที่ 4 ถ้าเป็นเท็จไปบรรทัดที่ 5

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "A" แล้วไปบรรทัดที่ 14

บรรทัดที่ 5 ปิดประโยคโครงสร้างเงื่อนไข if (เงื่อนไขที่ 1) และเริ่มโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 2) เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 70 จริงหรือไม่ ถ้าเป็นจริงทำคำสั่งบรรทัดที่ 6 ถ้าเป็นเท็จบรรทัดที่ 7

บรรทัดที่ 6 กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "B" แล้วไปทำคำสั่งบรรทัดที่ 14

บรรทัดที่ 7 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 2) และเริ่มโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 3) เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 60 จริงหรือไม่ ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 8 ถ้าเป็นเท็จไปทำคำสั่งบรรทัดที่ 9

บรรทัดที่ 8 กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "C" แล้วไปบรรทัดที่ 14

บรรทัดที่ 9 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 3) และเริ่มโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 4) เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 50 จริงหรือไม่ ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 10 ถ้าเป็นเท็จไปทำคำสั่งบรรทัดที่ 11

บรรทัดที่ 10 กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "D" แล้วไปบรรทัดที่ 14

บรรทัดที่ 11 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 4) และเริ่มโครงสร้างเงื่อนไข else หมายความว่าเป็นเท็จจากทุกกรณีก่อนหน้าจะไปทำคำสั่งบรรทัดที่ 12

บรรทัดที่ 12 กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "E" แล้วไปบรรทัดที่ 14

บรรทัดที่ 13 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 4)

บรรทัดที่ 14 แสดงข้อความ Score is 78: Grade evaluation is B.

บรรทัดที่ 15 สิ้นสุดสคริปต์ PHP

4.1.4 โครงสร้างเงื่อนไข switch (The switch Statement)

โครงสร้างเงื่อนไข switch เป็นโครงสร้างเงื่อนไขหนึ่งในโครงสร้างควบคุมขั้นพื้นฐาน เป็นการผสมผสานของ if ... else โดยโครงสร้างเงื่อนไข switch นิยมใช้สำหรับการเปรียบเทียบตัวแปรตัวเดียวแต่มีค่าของตัวแปรจำนวนมาก เพื่อกำหนดทางเลือกตามค่าของตัวแปร อาจจะถูกกำหนดเป็นค่าใดค่าหนึ่งจากตัวเลือกที่กำหนด

โครงสร้างเงื่อนไข switch เป็นโครงสร้างเงื่อนไขในการเปรียบเทียบค่าตัวแปรในกรณีต่างๆ ภายในโครงสร้างเงื่อนไข switch ถ้าตัวแปรนั้นๆ ตรงกับ case ไหนก็จะทำงานภายในโครงสร้างของกรณีนั้นๆ และจะหยุดการทำงานคำสั่งในแต่ละกรณี ด้วยคำสั่ง break (หากไม่มีคำสั่ง break ในแต่ละกรณีโปรแกรมจะทำการเปรียบเทียบเงื่อนไขลำดับถัดๆ ไป ทั้งหมด) แต่ถ้าค่าตัวแปรไม่ตรงกับ case ใดๆ เลยจะเข้าสู่การทำงานในส่วนของ case "default" มีรูปแบบ ดังนี้

รูปแบบ

```
switch (ตัวแปร) {
        case ค่าที่ 1 : ประโยคคำสั่งที่ 1;  break;
        case ค่าที่ 2 : ประโยคคำสั่งที่ 2;  break;
        ...
        case ค่าที่ N : ประโยคคำสั่งที่ N;  break;
        default: ประโยคคำสั่ง   //ในกรณีที่ไม่ตรงกับเงื่อนไขใดๆ ก่อนหน้า
}
```

ตัวอย่างที่ 4.4 การใช้โครงสร้างเงื่อนไข switch สำหรับการเปรียบเทียบค่าตัวแปรจำนวนมาก

```
1   <?php
2       $i=2;
```

```
3               switch ($i) {
4                       case 0: echo "i equals 0"; break;
5                       case 1: echo "i equals 1"; break;
6                       case 2: echo "i equals 2"; break;
7                       default: echo "i is not equal to 0, 1 or 2";
8               }
9       ?>
```

จากตัวอย่างที่ 4.4 การใช้โครงสร้างเงื่อนไข switch สำหรับการเปรียบเทียบค่าตัวแปรจำนวนมาก อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $i มีค่าเท่ากับ 2

บรรทัดที่ 3 ตรวจสอบค่าตัวแปรโดยใช้โครงสร้างเงื่อนไข switch เปรียบเทียบค่าตัวแปร $i เพื่อกำหนดทางเลือกตามค่าของตัวแปร

บรรทัดที่ 4 เปรียบเทียบค่าของตัวแปร $i มีค่าเท่ากับ 0 จริงหรือไม่ ถ้าจริงแสดงคำว่า "i equals 0" แล้วออกจากโครงสร้างเงื่อนไข switch บรรทัดที่ 9 ถ้าเป็นเท็จไปบรรทัดที่ 5

บรรทัดที่ 5 เปรียบเทียบค่าของตัวแปร $i มีค่าเท่ากับ 1 จริงหรือไม่ ถ้าเป็นจริงแสดงคำว่า "i equals 1" แล้วออกจากโครงสร้างเงื่อนไข switch บรรทัดที่ 9 ถ้าเป็นเท็จไปบรรทัดที่ 6

บรรทัดที่ 6 เปรียบเทียบค่าของตัวแปร $i มีค่าเท่ากับ 2 จริงหรือไม่ ถ้าเป็นจริงแสดงคำว่า "i equals 2" แล้วออกจากโครงสร้างเงื่อนไข switch บรรทัดที่ 9 ถ้าเป็นเท็จไปบรรทัดที่ 7

บรรทัดที่ 7 ในกรณีที่เป็นเท็จจากทุกกรณีข้างต้น ให้แสดงคำว่า "i is not equal to 0, 1 or 2"

บรรทัดที่ 8 ประโยคปิดโครงสร้างเงื่อนไข switch

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.5 การใช้โครงสร้างเงื่อนไข switch ประยุกต์จากตัวอย่างที่ 4.3

```
1       <?php
2               $score=78;
3               switch ($score) {
4                       case ($score>=80): $grade="A"; break;
5                       case ($score>=70): $grade="B"; break;
6                       case ($score>=60): $grade="C"; break;
7                       case ($score>=50): $grade="D"; break;
8                       default: $grade="E";
```

```
9               }
10              printf ("Score is %d: Grade evaluation is %s.", $score, $grade);
11      ?>
```

ผลลัพธ์

```
Score is 78: Grade evaluation is B.
```

จากตัวอย่างที่ 4.5 การใช้โครงสร้างเงื่อนไข switch ในการเปรียบเทียบหลายเงื่อนไข อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $score มีค่าเท่ากับ 78

บรรทัดที่ 3 โครงสร้างเงื่อนไข switch เปรียบเทียบค่าตัวแปร $score

บรรทัดที่ 4 เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 80 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "A" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10

บรรทัดที่ 5 เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 70 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "B" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10

บรรทัดที่ 6 เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 60 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "C" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10

บรรทัดที่ 7 เปรียบเทียบค่าตัวแปร $score มากกว่าหรือเท่ากับ 50 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "D" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10

บรรทัดที่ 8 ในกรณีเป็นเท็จจากทุกกรณีข้างต้น กำหนดค่าให้กับตัวแปร $grade มีค่าเท่ากับ "E"

บรรทัดที่ 9 ประโยคปิดโครงสร้างเงื่อนไข switch

บรรทัดที่ 10 แสดงข้อความ Score is 78: Grade evaluation is B.

บรรทัดที่ 11 สิ้นสุดสคริปต์ PHP

4.2 โครงสร้างเงื่อนไขทำซ้ำ

โครงสร้างเงื่อนไขทำซ้ำ เป็นหนึ่งในโครงสร้างควบคุม เพื่อควบคุมคำสั่งส่วนใดส่วนหนึ่งให้ทำงานซ้ำตามเงื่อนไขที่กำหนด เหตุผลสำคัญเพื่อลดจำนวนบรรทัดในการเขียนสคริปต์ และเพิ่มความคล่องตัวในการแก้ไข เมื่อต้องการปรับเปลี่ยนรูปแบบการแสดงผล เป็นต้น ตัวอย่างเช่น ต้องการแสดงผลเลข 1 ถึง 1000 หากเขียนสคริปต์โดยใช้คำสั่งแสดงผลปกติ อาจต้องใช้จำนวนบรรทัดมากกว่า 1000 บรรทัด เพื่อให้แสดงผลลัพธ์ตามต้องการ ทำให้เสียเวลาและขาดความคล่องตัว ดังนั้นจึงเป็นเหตุผลให้ใช้โครงสร้างเงื่อนไขทำซ้ำ เพื่อลดจำนวนบรรทัด เพิ่มความคล่องตัวในการเขียน และแก้ไขสคริปต์

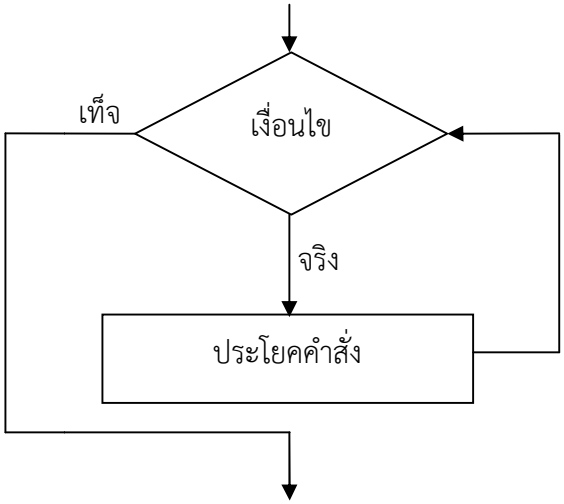
สำหรับในภาษา PHP มีโครงสร้างเงื่อนไขทำซ้ำที่สำคัญ ประกอบด้วย 1) while 2) do ... while 3) for และ 4) foreach มีรายละเอียดดังนี้

4.2.1 โครงสร้างเงื่อนไขทำซ้ำ while

โครงสร้างเงื่อนไขทำซ้ำ while เป็นหนึ่งในโครงสร้างเงื่อนไขทำซ้ำชนิดหนึ่ง หลักการทำงาน คือ ทำการตรวจสอบเงื่อนไขก่อนกระทำ ถ้าเงื่อนไขเป็นจริงจะเริ่มทำคำสั่งภายในโครงสร้างเงื่อนไขทำซ้ำ แล้วกลับมาตรวจสอบเงื่อนไขใหม่อีกครั้ง หากยังคงเป็นจริงอยู่ก็จะทำซ้ำเช่นนี้ไปเรื่อยๆ จนกระทั่งเงื่อนไขเป็นเท็จจึงจะออกจากโครงสร้างเงื่อนไขทำซ้ำได้ รูปแบบของโครงสร้างเงื่อนไขทำซ้ำ while มีดังนี้

รูปแบบ

```
while (เงื่อนไข) {
        ประโยคคำสั่ง
}
```



ภาพที่ 4.4 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ while

ตัวอย่างที่ 4.6 การใช้โครงสร้างเงื่อนไขทำซ้ำ while สำหรับแสดงตัวเลข 1 ถึง 5

```
1   <?php
2       $number = 1;
3       while ($number <= 5) {
4           echo $number
5           $number++;
6       }
7   ?>
```

ผลลัพธ์ คือ

```
1 2 3 4 5
```

จากตัวอย่างที่ 4.6 การใช้โครงสร้างเงื่อนไขทำซ้ำ while สำหรับแสดงตัวเลข 1 ถึง 5 อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $number มีค่าเท่ากับ 1 (ตัวแปรควบคุม)

บรรทัดที่ 3 เปรียบเทียบค่าตัวแปร $number มีค่าน้อยกว่าหรือเท่ากับ 5 จริงหรือไม่ ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 4 – 5 ถ้าเป็นเท็จไปบรรทัดที่ 7

บรรทัดที่ 4 แสดงค่าที่เก็บไว้ในตัวแปร $number

บรรทัดที่ 5 กำหนดให้ตัวแปร $number เพิ่มค่าอีก 1 ค่า (เพิ่มค่าตัวแปรควบคุม) แล้วกลับไปตรวจสอบเงื่อนไขในบรรทัดที่ 3

บรรทัดที่ 6 ประโยคปิดโครงสร้างเงื่อนไขทำซ้ำ while และกลับไปบรรทัดที่ 3

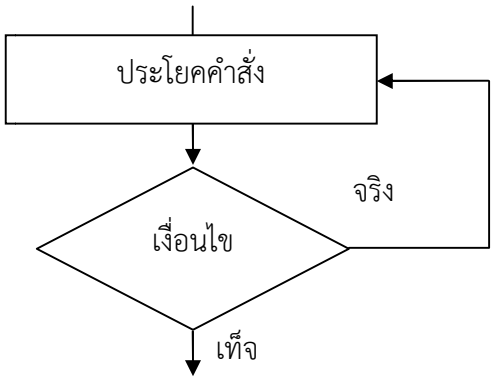
บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

4.2.2 โครงสร้างเงื่อนไขทำซ้ำ do ... while

โครงสร้างเงื่อนไขทำซ้ำ do ... while เป็นหนึ่งในโครงสร้างเงื่อนไขทำซ้ำชนิดหนึ่ง หลักการทำงาน คือ จะเริ่มทำคำสั่งภายในโครงสร้างเงื่อนไขทำซ้ำ do ... while ก่อน 1 ครั้ง แล้วจึงตรวจสอบเงื่อนไขภายหลัง ถ้าเงื่อนไขเป็นจริง โปรแกรมจะกลับมาทำซ้ำเช่นนี้ไปเรื่อยๆ จนกระทั่งเงื่อนไขเป็นเท็จจึงจะสามารถจบการทำงานภายในโครงสร้างเงื่อนไขทำซ้ำ do ... while ได้ โครงสร้างเงื่อนไขทำซ้ำ do ... while มีรูปแบบ ดังนี้

รูปแบบ

```
do{
        ประโยคคำสั่ง
} while (เงื่อนไข);
```



ภาพที่ 4.5 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ do ... while

ตัวอย่างที่ 4.7 การใช้โครงสร้างเงื่อนไขทำซ้ำ do ... while สำหรับแสดงตัวเลข 1 ถึง 5

```
1    <?php
2        $number = 0;
3        do {
4            $number++;
5            echo $number;
6        } while ($number < 5);
7    ?>
```

ผลลัพธ์ คือ

```
1 2 3 4 5
```

จากตัวอย่างที่ 4.7 การใช้โครงสร้างเงื่อนไขทำซ้ำ do ... while สำหรับแสดงตัวเลข 1 ถึง 5 อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $number มีค่าเท่ากับ 0 (ตัวแปรควบคุม)

บรรทัดที่ 3 เริ่มต้นโครงสร้างเงื่อนไขทำซ้ำ do ... while

บรรทัดที่ 4 กำหนดให้ตัวแปร $number เพิ่มค่าอีก 1 ค่า (เพิ่มค่าตัวแปรควบคุม)

บรรทัดที่ 5 แสดงค่าที่เก็บไว้ในตัวแปร $number

บรรทัดที่ 6 เปรียบเทียบค่าตัวแปร $number มีค่าน้อยกว่า 5 จริงหรือไม่ ถ้าเป็นจริงกลับไปทำคำสั่งบรรทัดที่ 4 ถ้าเป็นเท็จไปบรรทัดที่ 7

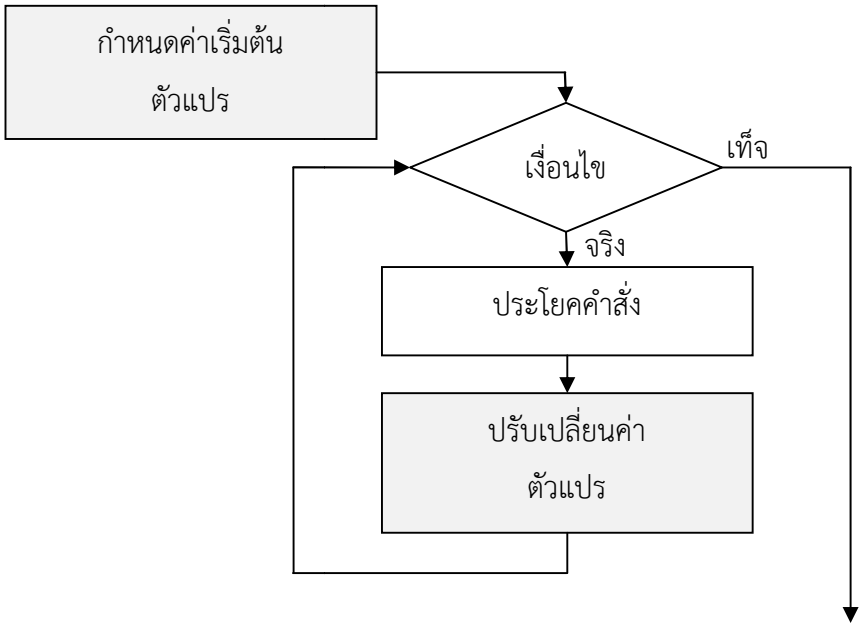
บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

4.2.3 โครงสร้างเงื่อนไขทำซ้ำ for

โครงสร้างเงื่อนไขทำซ้ำ for จัดเป็นโครงสร้างที่ใช้ในการควบคุมการทำงานของโปรแกรมประเภทโครงสร้างเงื่อนไขทำซ้ำ มีลักษณะคล้ายๆ กับโครงสร้างเงื่อนไขทำซ้ำ while ต่างที่โครงสร้างเงื่อนไขทำซ้ำ for สามารถกำหนดค่าเริ่มต้นตัวแปรเพื่อใช้เป็นตัวแปรควบคุมหรืออื่นๆ (กำหนดได้มากกว่า 1 ตัวแปร) ลักษณะรูปแบบที่สำคัญ แบ่งออกเป็น 3 ส่วน คือ 1) ส่วนกำหนดค่าเริ่มต้นตัวแปร 2) ส่วนของเงื่อนไข และ 3) ส่วนของการปรับเปลี่ยนค่าตัวแปร โดยทั้ง 3 ส่วน ถูกบรรจุอยู่ภายในโครงสร้างเงื่อนไขทำซ้ำ for เพียงบรรทัดเดียว มีรูปแบบดังนี้

รูปแบบ

```
for (ค่าเริ่มต้นตัวแปร; เงื่อนไข; ปรับเปลี่ยนค่าตัวแปร) {
    ประโยคคำสั่ง;
}
```



ภาพที่ 4.6 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ for

หมายเหตุ

จากรูปแบบ for (ค่าเริ่มต้นตัวแปร; เงื่อนไข; ปรับเปลี่ยนค่าตัวแปร) ค่าในแต่ละตำแหน่งสามารถเว้นเป็นค่าว่างได้ และใช้ตรรกศาสตร์อื่นๆ ในประโยคคำสั่ง เพื่อควบคุมการทำงานของโครงสร้างเงื่อนไขทำซ้ำ for

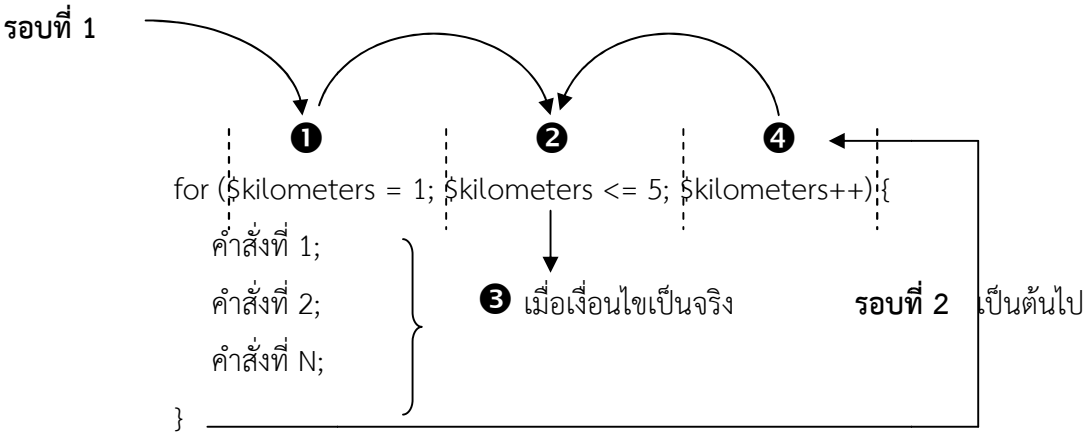
ตัวอย่างที่ 4.8 ตัวอย่างการใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบปกติ

```
1    <?php
2        for ($kilometers = 1; $kilometers <= 5; $kilometers++) {
3            printf ("%d Kilometers = ", $kilometers);
4            printf("%.5f Miles <br>", $kilometers * 0.62140);
5        }
6    ?>
```

จากตัวอย่างที่ 4.8 การใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบปกติ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 อธิบายเพิ่มเติม ดังนี้



ภาพที่ 4.7 แสดงลักษณ์การวนรอบของโครงสร้างเงื่อนไขทำซ้ำ for แบบปกติ

❶ หมายถึง ส่วนกำหนดค่าเริ่มต้นให้กับตัวแปร (กำหนดให้เพียงครั้งแรกเท่านั้น)

❷ หมายถึง ส่วนตรวจสอบเงื่อนไข ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 3 ถ้าเป็นเท็จไปทำคำสั่งบรรทัดที่ 6

❸ หมายถึง เมื่อเงื่อนไขเป็นจริงให้ทำคำสั่งอะไรบ้าง เรียงตามลำดับ

❹ หมายถึง ส่วนปรับเปลี่ยนค่าให้กับตัวแปร

รอบที่ 1 จะเริ่มต้นที่ ❶ (กำหนดให้เพียงครั้งแรกเท่านั้น) โดยในรอบที่ 1 กำหนดให้ตัวแปร $kilometers มีค่าเท่ากับ 1 แล้วตรวจสอบเงื่อนไข ใน ❷ ตัวแปร $kilometers มีค่าน้อยกว่าหรือเท่ากับ 5 จริงหรือไม่ ถ้าเป็นจริง ให้ไปทำคำสั่งในบรรทัดที่ 3 และอื่นๆ เรียงตามลำดับคำสั่ง ถ้าเป็นเท็จไปบรรทัดที่ 6 (ออกจากโครงสร้างเงื่อนไขทำซ้ำ for)

รอบที่ 2 เป็นต้นไป จะกลับมาที่ ❹ เพื่อเพิ่มค่าให้กับตัวแปร $kilometers ครั้งละ 1 ค่า แล้วตรวจสอบเงื่อนไข ใน ❷ ตัวแปร $kilometers ยังคงมีค่าน้อยกว่าหรือเท่ากับ 5 จริงหรือไม่ ถ้าเป็นจริง ให้ไปทำคำสั่งบรรทัดที่ 3 – 5 ถ้าเป็นเท็จ ไปบรรทัดที่ 6 (ออกจากโครงสร้างเงื่อนไขทำซ้ำ for)

บรรทัดที่ 3 แสดงผลค่าตัวแปร $kilometers ในแต่ละรอบการทำซ้ำ และข้อความคำว่า Kilometers =

บรรทัดที่ 4 แสดงผลการคำนวณค่าตัวแปร $kilometers คูณด้วย 0.62140 ในแต่ละรอบการทำซ้ำ ข้อความคำว่า Miles แล้วขึ้นบรรทัดใหม่

บรรทัดที่ 5 ประโยคปิดโครงสร้างเงื่อนไขทำซ้ำ for และกลับไปส่วนที่ ❹ ในบรรทัดที่ 2

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.9 ตัวอย่างการใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดเงื่อนไข

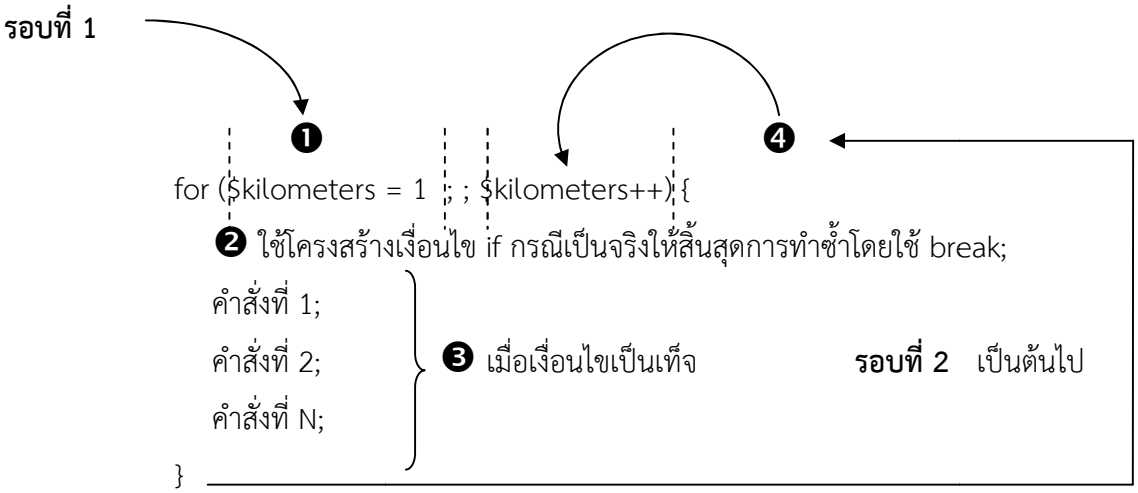
```
1       <?php
2               for ($kilometers = 1; ; $kilometers++) {
3                       If ($kilometers > 5) break;
4                       printf ("%d Kilometers = ", $kilometers);
```

```
5                   printf("%.5f Miles <br>", $kilometers * 0.62140);
6               }
7       ?>
```

จากตัวอย่างที่ 4.9 การใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดเงื่อนไข อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 อธิบายเพิ่มเติม ดังนี้



ภาพที่ 4.8 แสดงลักษณ์การวนรอบของโครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดเงื่อนไข

❶ หมายถึง ส่วนกำหนดค่าเริ่มต้นให้กับตัวแปร (กำหนดให้เพียงครั้งแรกเท่านั้น)

❷ หมายถึง ส่วนตรวจสอบเงื่อนไข ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 4 ถ้าเป็นเท็จไปทำคำสั่งบรรทัดที่ 7

❸ หมายถึง ทำคำสั่งเรียงตามลำดับ (อยู่ภายใต้โครงสร้างเงื่อนไขทำซ้ำ for)

❹ หมายถึง ส่วนปรับเปลี่ยนค่าให้กับตัวแปร

รอบที่ 1 จะเริ่มต้นที่ ❶ (กำหนดให้เพียงครั้งแรกเท่านั้น) โดยในรอบที่ 1 กำหนดให้ตัวแปร $kilometers มีค่าเท่ากับ 1 แล้วไปทำคำสั่งในบรรทัดที่ 3

รอบที่ 2 เป็นต้นไป จะกลับมาที่ ❹ เพื่อเพิ่มค่าให้กับตัวแปร $kilometers ครั้งละ 1 ค่า แล้วไปทำคำสั่งในบรรทัดที่ 3

บรรทัดที่ 3 ตรวจสอบเงื่อนไขตัวแปร $kilometers มีค่ามากกว่า 5 จริงหรือไม่ เมื่อตรวจสอบเงื่อนไขแล้ว ถ้าเป็นจริงให้หยุดและออกจากโครงสร้างเงื่อนไขทำซ้ำ for ไปบรรทัดที่ 7 ถ้าเป็นเท็จไปทำคำสั่งบรรทัดที่ 4

บรรทัดที่ 4 แสดงผลค่าตัวแปร $kilometers ในแต่ละรอบการทำซ้ำ และข้อความคำว่า Kilometers =

บรรทัดที่ 5 แสดงผลการคำนวณค่าตัวแปร $kilometers คูณด้วย 0.62140 ในแต่ละรอบการทำซ้ำ ข้อความคำว่า Miles แล้วขึ้นบรรทัดใหม่

บรรทัดที่ 6 ประโยคปิดโครงสร้างเงื่อนไขทำซ้ำ for และกลับไปส่วนที่ ❹ ในบรรทัดที่ 2

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.10 ตัวอย่างการใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดค่าใดๆ

```
1   <?php
2       $kilometers = 1;
3       for (; ;) {
4           If ($kilometers > 5) break;
5           printf ("%d Kilometers = ", $kilometers);
6           printf("%.5f Miles <br>", $kilometers * 0.62140);
7           $kilometers++
8       }
9   ?>
```

จากตัวอย่างที่ 4.10 การใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดค่าใดๆ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $kilometers มีค่าเท่ากับ 1

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดค่าใดๆ (วนรอบโดยไม่มีเงื่อนไขการสิ้นสุด ลักษณะการทำงานดังกล่าวผู้เขียนโปรแกรมจะต้องระมัดระวัง เนื่องจากอาจทำให้โปรแกรมวนรอบไม่รู้จบ)

บรรทัดที่ 4 ตรวจสอบเงื่อนไขว่า ตัวแปร $kilometers มีค่ามากกว่า 5 จริงหรือไม่ ถ้าเป็นจริง ให้หยุดและออกจากโครงสร้างเงื่อนไขทำซ้ำ for ไปบรรทัดที่ 9 ถ้าเป็นเท็จไปทำคำสั่งบรรทัดที่ 5

บรรทัดที่ 5 แสดงผลค่าตัวแปร $kilometers ในแต่ละรอบการทำซ้ำ และข้อความคำว่า Kilometers =

บรรทัดที่ 6 แสดงผลการคำนวณค่าตัวแปร $kilometers คูณด้วย 0.62140 ในแต่ละรอบการทำซ้ำ ข้อความคำว่า Miles แล้วขึ้นบรรทัดใหม่

บรรทัดที่ 7 เพิ่มค่าให้กับตัวแปร $kilometers ครั้งละ 1 ค่า

บรรทัดที่ 8 ประโยคปิดโครงสร้างเงื่อนไขทำซ้ำ for และกลับไปทำคำสั่งบรรทัดที่ 4

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

ผลลัพธ์ทั้ง 3 ตัวอย่าง คือ

```
1 kilometers = 0.6214 miles
2 kilometers = 1.2428 miles
```

```
3 kilometers = 1.8642 miles
4 kilometers = 2.4856 miles
5 kilometers = 3.107 miles
```

4.2.4 โครงสร้างทำซ้ำ foreach

โครงสร้างทำซ้ำ foreach ใช้สำหรับการทำงานร่วมกับข้อมูลชนิดอาร์เรย์เท่านั้น (การจัดการกับข้อมูลชนิดอาร์เรย์ จะกล่าวถึงในบทถัดไป) ช่วยในการทำซ้ำหรือวนรอบดึงค่าข้อมูลอาร์เรย์ออกจากตัวแปรชนิดอาร์เรย์ตามลำดับ รูปแบบของโครงสร้างทำซ้ำ foreach มีรูปแบบการใช้งาน 2 รูปแบบ ดังนี้

รูปแบบที่ 1

```
foreach (array_expression as $value) {
        ประโยคคำสั่ง
}
```

เมื่อ array_expression หมายถึง ตัวแปรชนิดอาร์เรย์ที่ต้องการอ่านค่า
$value หมายถึง ตัวแปรที่ใช้รับผลจากการอ่านค่าข้อมูลชนิดอาร์เรย์

รูปแบบที่ 2

```
foreach (array_expression as $key => $value)
        ประโยคคำสั่ง
?>
```

เมื่อ array_expression หมายถึง ตัวแปรชนิดอาร์เรย์ที่ต้องการอ่านค่า
$key หมายถึง ลำดับคีย์ที่ต้องการอ่านค่า
$value หมายถึง ตัวแปรที่ใช้รับผลจากการอ่านค่าข้อมูลชนิดอาร์เรย์

รูปแบบที่ 1 โครงสร้างทำซ้ำ foreach จะทำการวนรอบเพื่ออ่านค่าข้อมูลของสมาชิกในอาร์เรย์ โดยแต่ละรอบนั้น ค่าของสมาชิกปัจจุบันจะถูกกำหนดไปยังตัวแปร $value และตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ (array pointer) จะเลื่อนไปยังสมาชิกลำดับถัดไปเพื่อรอการวนรอบใหม่ (ดังจะเห็นได้จากเมื่อเริ่มการวนรอบใหม่ ค่าของสมาชิกลำดับถัดไปจะถูกกำหนดไปยัง $value แทน) โดยจะทำการวนรอบอย่างนี้ไปเรื่อยๆ จนกระทั่งครบจำนวนสมาชิกทั้งหมดในอาร์เรย์

รูปแบบที่ 2 โครงสร้างทำซ้ำ foreach จะทำงานคล้ายกับรูปแบบที่ 1 แต่ค่าที่ส่งนั้นจะประกอบด้วย คีย์ และค่า ของสมาชิกแต่ละตัวในอาร์เรย์ ซึ่งรูปแบบที่ 1 นั้นจะทำการส่งเฉพาะค่าของสมาชิกเพียงอย่างเดียว

เมื่อเริ่มการวนรอบในครั้งแรกโครงสร้างทำซ้ำ foreach จะกำหนดให้ตัวชี้ตำแหน่งสมาชิกในอาร์เรย์เริ่มต้นชี้ไปยังตำแหน่งของสมาชิกลำดับแรกโดยอัตโนมัติ หมายความว่าไม่จำเป็นต้องใช้ฟังก์ชัน reset ( ) ก่อนเข้าการวนรอบในโครงสร้างทำซ้ำ foreach และยังสามารถเปลี่ยนแปลงค่าสมาชิกในอาร์เรย์ได้

ตัวอย่างที่ 4.11 การใช้โครงสร้างทำซ้ำ foreach

```
1    <?php
2        $arr = array (1, 2, 3, 4);
3        foreach ($arr as $value) {
4            printf ("%d ", $value);
5        }
6    ?>
```

ผลลัพธ์

```
1 2 3 4
```

จากตัวอย่างที่ 4.11 ตัวอย่างการใช้โครงสร้างทำซ้ำ foreach นิยมใช้เพื่อแสดงผลข้อมูลในอาร์เรย์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $arr เป็นชนิดอาร์เรย์ ประกอบด้วยค่า 1, 2, 3 และ 4 ตามลำดับ

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ foreach

รอบที่ 1 จะกำหนดให้ตัวชี้ตำแหน่งอาร์เรย์ ชี้ไปยังตำแหน่งสมาชิกอาร์เรย์ลำดับที่ 1 ในตัวแปร $arr อ่านค่าข้อมูลจากสมาชิกอาร์เรย์ ลำดับที่ 1 แล้วกำหนดค่าที่อ่านได้ไปยังตัวแปร $value หลังจากนั้นไปทำคำสั่งในบรรทัดที่ 4

รอบถัดไป ทำการเลื่อนตัวชี้ตำแหน่งอาร์เรย์ไปยังลำดับสมาชิกอาร์เรย์ลำดับถัดไปในตัวแปร $arr อ่านค่าข้อมูลสมาชิกอาร์เรย์ แล้วกำหนดค่าที่อ่านได้ไปยังตัวแปร $value หลังจากนั้นไปทำคำสั่งในบรรทัดที่ 4 หากไม่สามารถอ่านค่าข้อมูลได้ แสดงว่าข้อมูลของสมาชิกอาร์เรย์ในตัวแปร $arr ได้ถูกอ่านครบแล้ว จะออกจากโครงสร้างทำซ้ำ foreach ไปบรรทัดที่ 6

บรรทัดที่ 4 แสดงค่าที่เก็บไว้ในตัวแปร $value

บรรทัดที่ 5 ประโยคปิดโครงสร้างทำซ้ำ foreach และกลับไปทำคำสั่งบรรทัดที่ 3

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

4.3 คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ

คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ คือ คำสั่งควบคุมที่ส่งผลต่อการทำงานของโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ เช่น ในบางครั้งของการเขียนโปรแกรมอาจจำเป็นต้องหยุดการทำงานของโครงสร้างเงื่อนไขทำซ้ำนั้นๆ อันเนื่องมาจากค่าตัวแปรบางอย่างอาจก่อให้เกิดข้อผิดพลาดจำเป็นต้องหยุดการทำงานก่อนปัญหาดังกล่าวจะเกิดขึ้น เป็นต้น คำสั่ง

ควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ ประกอบด้วย 1) คำสั่ง break และ 2) คำสั่ง continue มีรายละเอียด ดังนี้

4.3.1 คำสั่ง break (The break Statements)

คำสั่ง break เป็นคำสั่งที่สั่งให้หยุดการทำงานโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ ไม่ว่าจะเป็น for, foreach, while, do ... while หรือ switch ก็ตาม เมื่อโปรแกรมมาพบคำสั่ง break โปรแกรมจะหยุดประมวลผลคำสั่งทั้งหมดในโครงสร้างเงื่อนไขหรือโครงสร้างเงื่อนไขทำซ้ำ แล้วจะกระโดดออกจากโครงสร้างเงื่อนไขดังกล่าวทันที และทำต่อในลำดับบรรทัดถัดไป (หากมี)

คำสั่ง break สามารถกำหนดพารามิเตอร์ โดยการระบุหมายเลขต่อท้ายคำสั่ง ประกอบด้วย 1 หมายถึง การสั่งให้ออกจากการทำงานโครงสร้างเงื่อนไข ชั้นที่ 1 (ค่าโดยปริยาย หากไม่ระบุจะมีค่าเท่ากับ 1) และ 2 หมายถึง การสั่งให้ออกจากโครงสร้างเงื่อนไข ชั้นที่ 1 และ 2 ตัวอย่างดังนี้

ตัวอย่างที่ 4.12 การใช้คำสั่ง break สำหรับออกจากการทำงานของโครงสร้างเงื่อนไขทำซ้ำ

```
1   <?php
2       $arr = array ("one", "two", "three", "four", "stop", "five");
3       foreach ($arr as $value) {
4           if ($value == "stop") break;
5           echo "$value ";
6       }
7   ?>
```

ผลลัพธ์

```
one two three four
```

จากตัวอย่างที่ 4.12 การใช้คำสั่ง break สำหรับออกจากการทำงานของโครงสร้างเงื่อนไขทำซ้ำ foreach เมื่อค่าของอาร์เรย์มีค่าเท่ากับ "stop" อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $arr เป็นชนิดอาร์เรย์ ประกอบด้วยค่า "one", "two", "three", "four", "stop" และ "five" ตามลำดับ

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ foreach

รอบที่ 1 จะกำหนดให้ตัวชี้ตำแหน่งอาร์เรย์ ชี้ไปยังตำแหน่งสมาชิกอาร์เรย์ลำดับที่ 1 ในตัวแปร $arr อ่านค่าข้อมูลจากสมาชิกอาร์เรย์ ลำดับที่ 1 แล้วกำหนดค่าที่อ่านได้ไปยังตัวแปร $value หลังจากนั้นไปทำคำสั่งในบรรทัดที่ 4

รอบถัดไป ทำการเลื่อนตัวชี้ตำแหน่งอาร์เรย์ไปยังลำดับสมาชิกอาร์เรย์ลำดับถัดไปในตัวแปร $arr อ่านค่าข้อมูลสมาชิกอาร์เรย์ แล้วกำหนดค่าที่อ่านได้ไปยังตัวแปร $value

หลังจากนั้นไปทำคำสั่งในบรรทัดที่ 4 หากไม่สามารถอ่านค่าข้อมูลได้ แสดงว่าข้อมูลของสมาชิกอาร์เรย์ในตัวแปร $arr ได้ถูกอ่านครบแล้ว จะออกจากโครงสร้างทำซ้ำ foreach ไปบรรทัดที่ 7

บรรทัดที่ 4 โครงสร้างเงื่อนไขตรวจสอบค่าที่เก็บในตัวแปร $value หากมีค่าเท่ากับ "stop" ใช้คำสั่ง break เพื่อออกจากโครงสร้างเงื่อนไขทำซ้ำ foreach ไปยังบรรทัดที่ 7

บรรทัดที่ 5 แสดงค่าที่เก็บไว้ในตัวแปร $value

บรรทัดที่ 6 ประโยคปิดโครงสร้างทำซ้ำ foreach และกลับไปทำคำสั่งบรรทัดที่ 3

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.13 การใช้คำสั่ง break กำหนดพารามิเตอร์เพื่อออกจากโครงสร้างเงื่อนไขทำซ้ำ

```
1   <?php
2       $number = 1;
3       while ($number<=20) {
4           switch ($number) {
5               case 5: echo "Number = 5 <br>"; break 1;
6               case 10: echo "Number = 10 Quitting"; break 2;
7               default: break;
8           }
9           $number++;
10      }
11  ?>
```

ผลลัพธ์

```
Number = 5
Number = 10 Quitting
```

จากตัวอย่างที่ 4.13 การใช้คำสั่ง break กำหนดพารามิเตอร์เพื่อออกจากโครงสร้างเงื่อนไขทำซ้ำ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $number มีค่าเท่ากับ 1

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ while ตรวจสอบค่าตัวแปร $number มีค่าน้อยกว่าหรือเท่ากับ 20 จริงหรือไม่ หากจริงไปทำคำสั่งในบรรทัดที่ 4 หากเป็นเท็จไปบรรทัด 11

บรรทัดที่ 4 โครงสร้างเงื่อนไข switch ตรวจสอบค่าตัวแปร $number

บรรทัดที่ 5 กรณีค่าของตัวแปร $number มีค่าเท่ากับ 5 ให้แสดงข้อความ "Number = 5" หลังจากนั้นขึ้นบรรทัดใหม่
 และออกจากโครงสร้างเงื่อนไข switch ไปทำคำสั่งบรรทัดที่ 9 แสดงค่าที่เก็บไว้ในตัวแปร $value

บรรทัดที่ 6 กรณีค่าของตัวแปร $number มีค่าเท่ากับ 10 ให้แสดงข้อความ " Number = 10 Quitting" แล้วออกจากโครงสร้างเงื่อนไขทำซ้ำ while ไปบรรทัดที่ 11

บรรทัดที่ 7 กรณีอื่นๆ ให้ออกจากโครงสร้างเงื่อนไข switch

บรรทัดที่ 8 ปิดประโยคโครงสร้างเงื่อนไข switch

บรรทัดที่ 9 กำหนดให้ตัวแปร $number เพิ่มค่า 1 ค่า

บรรทัดที่ 10 ปิดประโยคโครงสร้างเงื่อนไขทำซ้ำ while

บรรทัดที่ 11 สิ้นสุดสคริปต์ PHP

4.3.2 คำสั่ง continue (The continue Statement)

คำสั่ง continue นิยมใช้ภายในโครงสร้างเงื่อนไขทำซ้ำ เพื่อให้ข้ามการวนรอบของรอบการทำงานโครงสร้างเงื่อนไขทำซ้ำในปัจจุบัน ไปทำงานในวนรอบถัดไป และคำสั่ง continue ยังสามารถกำหนดพารามิเตอร์ เพื่อข้ามการทำงานในวนรอบระดับ (level) ชั้นใดก็ได้ ตัวอย่างดังนี้

ตัวอย่างที่ 4.14 การใช้คำสั่ง continue ข้ามการทำงานวนรอบของโครงสร้างเงื่อนไขทำซ้ำปัจจุบัน ไปทำงานในวนรอบถัดไป

```
1	<?php
2		$number = 0;
3		while ($number <= 20) {
4			$number++;
5			if ($number%2) continue;
6			echo "$number ";
7		}
8	?>
```

ผลลัพธ์

```
	2 4 6 8 10 12 14 16 18 20
```

จากตัวอย่างที่ 4.14 การใช้คำสั่ง continue ข้ามการทำงานวนรอบของโครงสร้างเงื่อนไขทำซ้ำปัจจุบัน ไปทำงานในวนรอบถัดไป อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร $number มีค่าเท่ากับ 0

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ while ตรวจสอบค่าตัวแปร $number มีค่าน้อยกว่าหรือเท่ากับ 20 จริงหรือไม่ หากจริงไปทำคำสั่งในบรรทัดที่ 4 หากเป็นเท็จไปบรรทัด 8

บรรทัดที่ 4 เพิ่มค่าให้กับตัวแปร $number 1 ค่า

บรรทัดที่ 5 โครงสร้างเงื่อนไข if ตรวจสอบผลการคำนวณของนิพจน์เพื่อหารหาผลเศษของตัวแปร $number ด้วย 2 ถ้าผลหารมีค่าเป็น 1 (จริง) คือ หารไม่ลงตัว ให้ทำคำสั่ง continue คือ ข้ามการทำงานวนรอบปัจจุบัน กลับไปตรวจสอบเงื่อนไขบรรทัดที่ 3

บรรทัดที่ 6 แสดงค่าตัวแปร $number (ผลลัพธ์ที่ได้ คือ แสดงผลเลขคู่)

บรรทัดที่ 7 ปิดประโยคโครงสร้างเงื่อนไขทำซ้ำ while

บรรทัดที่ 8 สิ้นสุดสคริปต์ PHP

สรุป

โครงสร้างควบคุมเป็นส่วนที่สำคัญมากสำหรับการควบคุมการไหลของสคริปต์คำสั่ง การกำหนดทางเลือก ตรวจสอบเงื่อนไข การทำคำสั่งซ้ำๆ ตามเงื่อนไขกำหนด ในภาษา PHP มีโครงสร้างควบคุมที่สำคัญแบ่งออกเป็น 3 กลุ่ม ประกอบด้วย 1) โครงสร้างเงื่อนไข 2) โครงสร้างเงื่อนไขทำซ้ำ และ 3) คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข โดยก่อนเริ่มต้นการใช้โครงสร้างควบคุมนั้นจำเป็นต้องออกแบบและวิเคราะห์หาความต้องการก่อน หรืออาจเขียนเป็นแผนภาพการไหลของสคริปต์คำสั่งเพื่อจำลองลำดับเหตุการณ์ที่ต้องการให้ภาษา PHP ประมวลผล แล้วหลังจากนั้นพิจารณาเลือกใช้โครงสร้างควบคุมที่เหมาะสม จุดสำคัญของโครงสร้างควบคุมก็คือการเปรียบเทียบเงื่อนไขโดยใช้งานร่วมกับตัวดำเนินการเปรียบเทียบทางตรรกะ

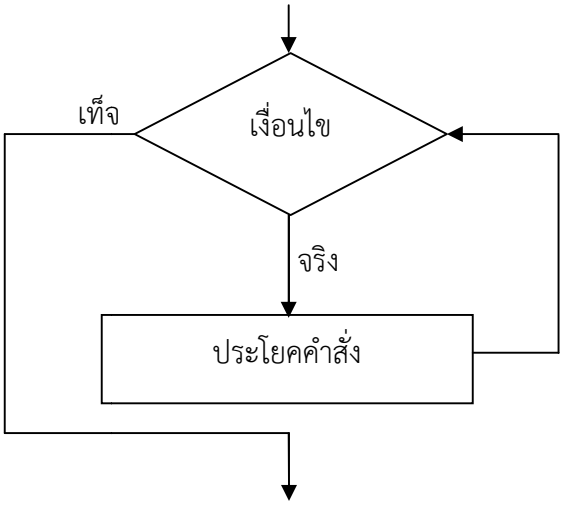
คำถามท้ายบท

1. โครงสร้างควบคุมแบ่งได้เป็นกี่กลุ่ม และประกอบด้วยกลุ่มอะไรบ้าง
2. จงอธิบายถึงรูปแบบโครงสร้างเงื่อนไข if และวาดแผนภาพการไหลของโครงสร้างเงื่อนไข if
3. จงอธิบายถึงรูปแบบโครงสร้างเงื่อนไขทำซ้ำ while และวาดแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ while
4. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย พร้อมผลลัพธ์ที่ได้จากสคริปต์

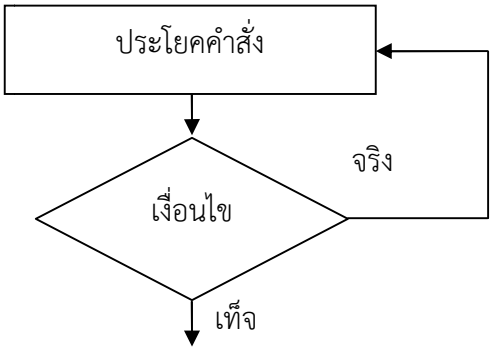
```
1    <?php
2        $row = 1;
3        $number = 12;
4        while ($row <= 20) {
5            printf ("%d %s %d %s %d <br>", $number, "x", $row, "=", $number*$row);
6            $row++;
7        }
8    ?>
```

5. จงอธิบายหลักการทำงานของโครงสร้างควบคุม ดังนี้

5.1 จากแผนภาพแสดงโครงสร้างควบคุมอะไร และมีหลักการทำงานอย่างไร



5.2 จากแผนภาพแสดงโครงสร้างควบคุมอะไร และมีหลักการทำงานอย่างไร



5.3 จากแผนภาพแสดงโครงสร้างควบคุมอะไร และมีหลักการทำงานอย่างไร