

เริ่มต้นการเขียนโปรแกรมด้วย ภาษา PHP

ใน Ebook เล่มนี้ ผู้เขียนจะแนะนำถึง พื้นฐานสำหรับผู้เริ่มต้นที่ต้องการ เริ่มต้นพัฒนา โปรแกรม ด้วยภาษา PHP และ จะสอดแทรกเทคนิค ต่าง ๆ ตัวอย่างการใช้งานจริงเพื่อเสริมสร้างความเข้าใจมากยิ่งขึ้น ทั้งนี้ใน Ebook เล่มนี้ นำเสนอ คำสั่งพื้นฐาน และ การใช้งานเบื้องต้นเท่านั้น ในส่วนท้ายจะมี ตัวอย่างการโปรแกรม เพิ่ม แก้ไข ลบข้อมูล ในฐานข้อมูล MySQL เพื่อเป็นแนวทางในการเขียนโปรแกรม

วิทยา ทองคำ

Monkeywebstuido.com

เนื้อหาจะมีอะไรบ้าง

- PHP คืออะไร
- การเตรียมเครื่องมือในการพัฒนาโปรแกรม
- การเขียน Tag PHP
- คำสั่งการแสดงผล
- การเขียน Comment ในรูปแบบต่าง ๆ
- การใช้งานคำสั่ง PHP ร่วมกับ ภาษา HTML
- ตัวแปร
- ตัวดำเนินการ
- คำสั่งตรวจสอบเงื่อนไข
- คำสั่งวนรอบการทำงาน
- Array
- Function
- PHP MySQLi
- การเชื่อมต่อฐานข้อมูล
- การเพิ่มข้อมูล
- การแก้ไขข้อมูล
- การลบข้อมูล

PHP คืออะไร

PHP คือ ภาษาโปรแกรม รูปแบบหนึ่ง ที่ทำงาน อยู่บน Server หรือ เรียกว่า Server Side Script โดยคำสั่ง ภาษา PHP นั้นจะถูกนำไปประมวลผล ในฝั่ง Server ก่อน และส่งผลลัพธ์ กลับมาแสดงผลให้ ผู้ใช้งานผ่านทางหน้าจอของ เครื่อง Client ที่ร้องขอไฟล์

ภาษา PHP นั้นถูกพัฒนาอย่างต่อเนื่อง มี library และ Framework ต่าง ๆ ให้เลือกใช้มากมาย เป็น ภาษาที่ได้รับความนิยม มีผู้ใช้งานเป็นจำนวนมาก เหมาะสำหรับเริ่มต้นเรียนรู้การเขียนโปรแกรม

เครื่องมือสำหรับการพัฒนาโปรแกรม ภาษา PHP

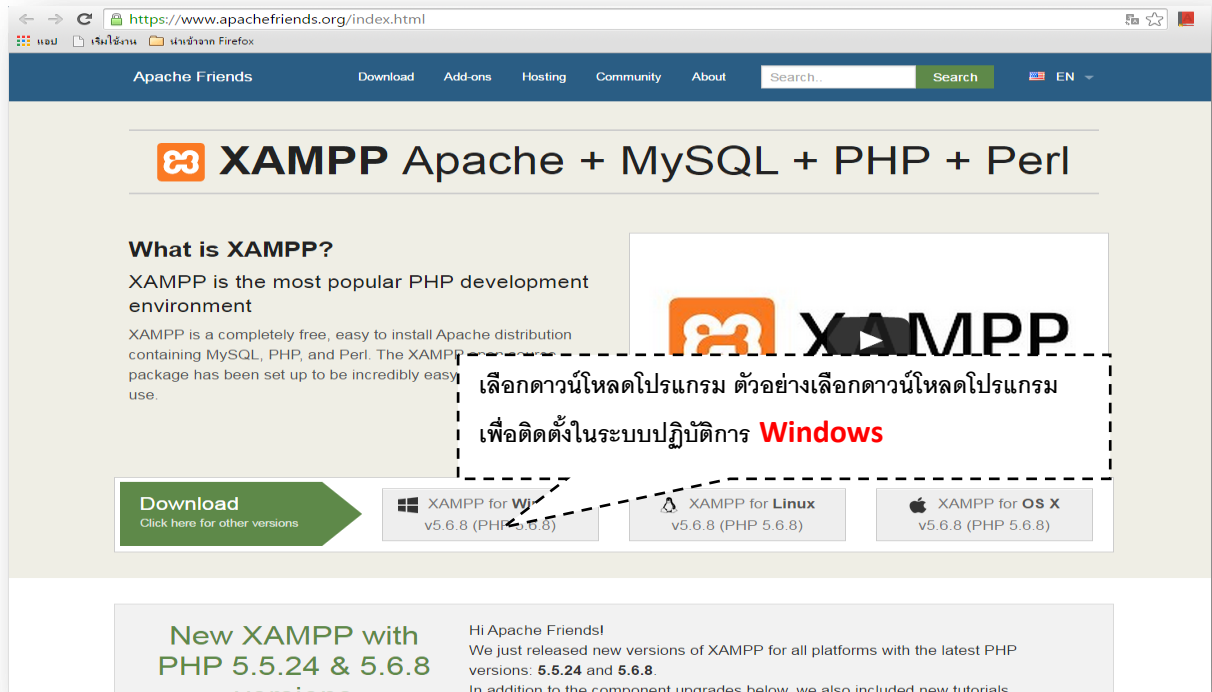
1. โปรแกรม Text Editor เช่น notepad ++ , sublime text 3 , netbeans เป็นต้น
2. โปรแกรมจำลอง Web Server เช่น Xampp , Wampp เป็นต้น โดยโปรแกรมจำลอง Web Server นั้น คือ โปรแกรมที่ รวมเอาโปรแกรมที่จำเป็นในการพัฒนา เช่น Apache , MySQL , PhpMyAdmin และ อื่น ๆ เพื่อให้สามารถพัฒนาโปรแกรมได้สะดวกยิ่งขึ้น
3. โปรแกรม Web Browser Google Chrome หรือ Mozilla Firefox

ใน Ebook เล่มนี้ จะใช้ โปรแกรม Text Editor Sublime Text และ โปรแกรมจำลอง Web Server Xampp

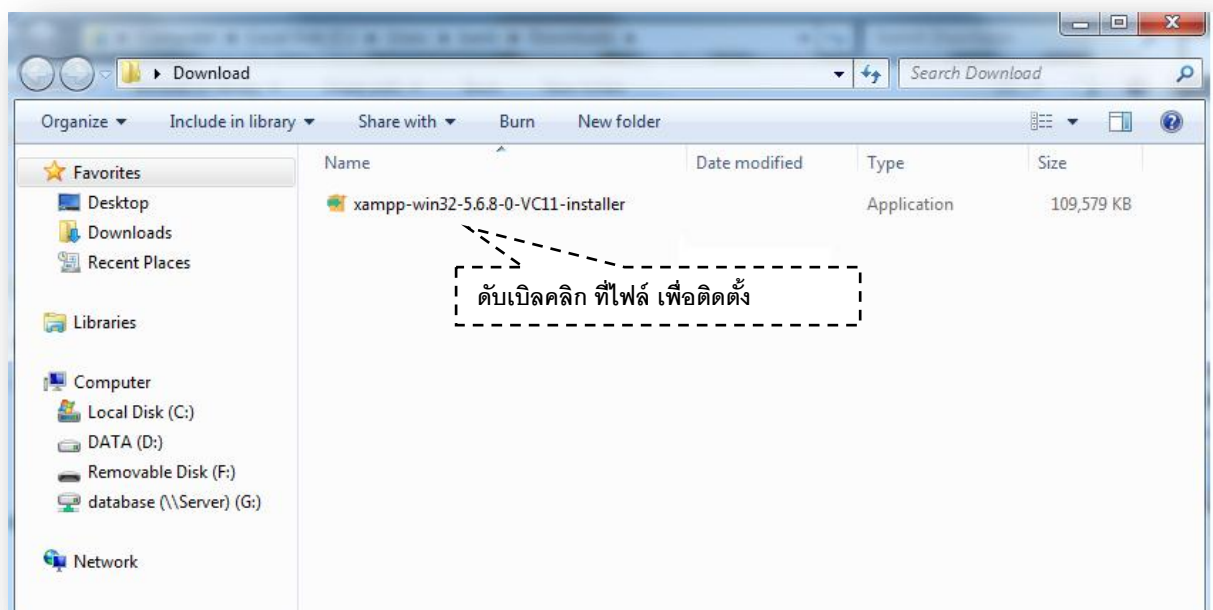
ในการเริ่มต้นผู้เขียนแนะนำให้ ผู้อ่านได้ ทดลองพิมพ์คำสั่ง และ Run ตาม เพื่อให้เกิดความคุ้นเคยกับตัว ภาษาโปรแกรม

การดาวน์โหลดและติดตั้ง โปรแกรมจำลอง Server Xampp (For Windows)

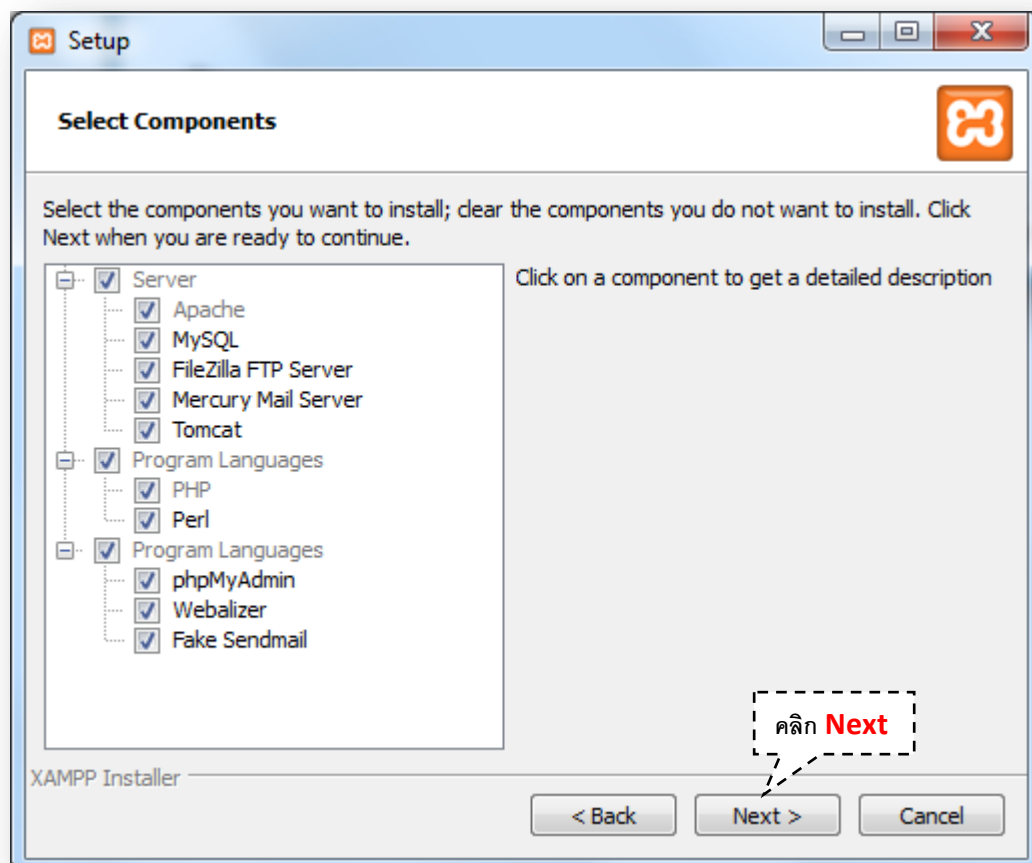
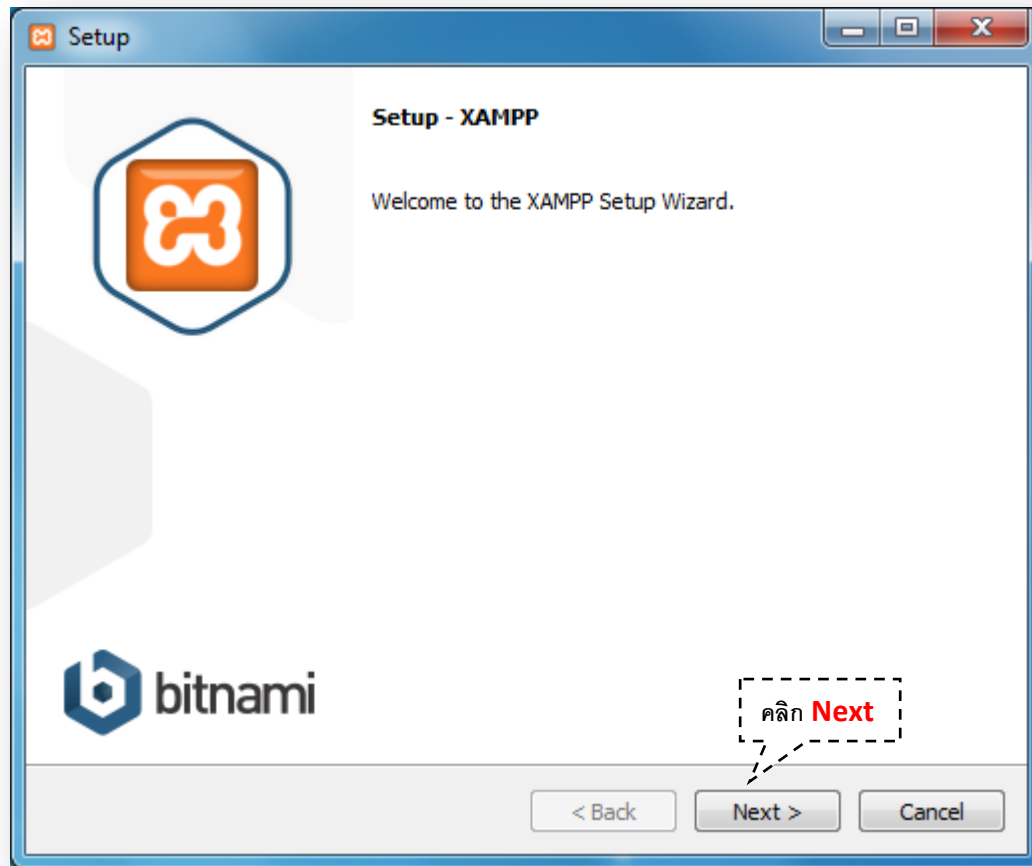
- โปรแกรม Xampp เป็นโปรแกรมสำหรับจำลองเครื่องคอมพิวเตอร์ให้สามารถใช้งาน Web Server ได้สามารถดาวน์โหลด โปรแกรมได้ที่ www.apachefriends.org
- โปรดตรวจสอบว่าในเครื่องของท่านไม่มีโปรแกรม Web Server ตัวอื่น ใช้งานอยู่



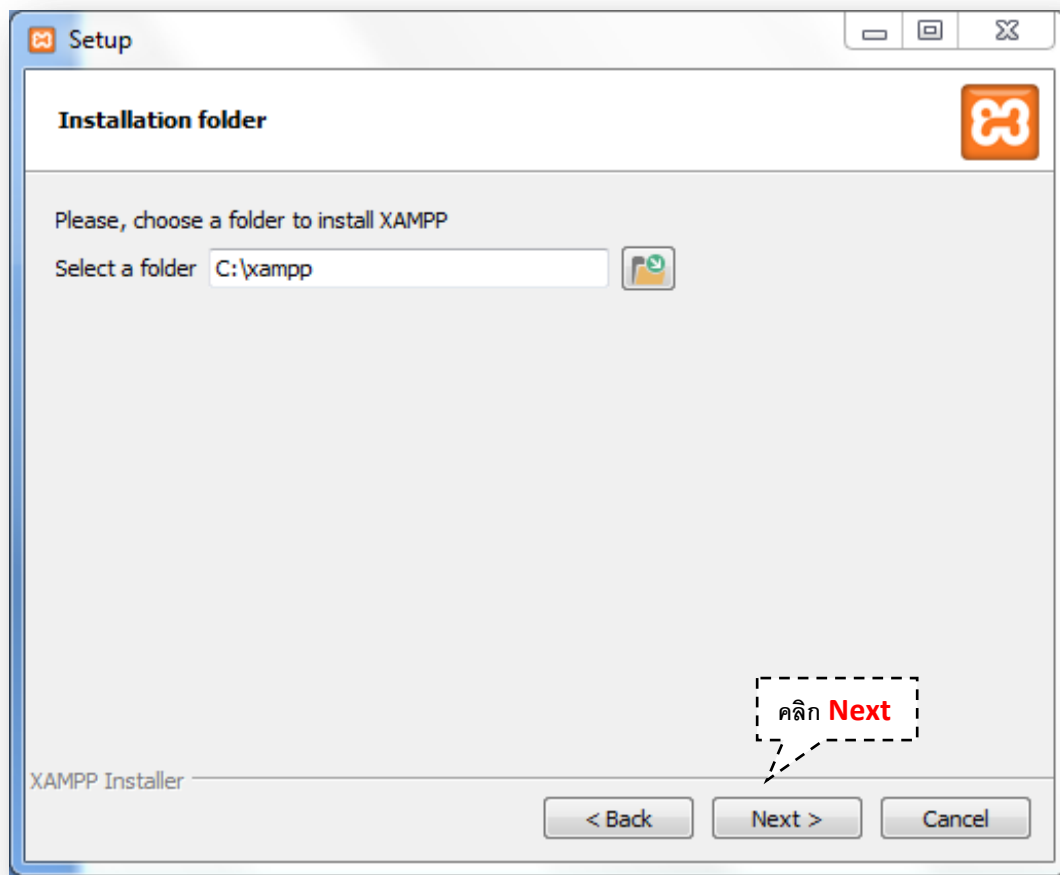
- เมื่อดาวน์โหลดเรียบร้อยแล้วจะได้ไฟล์ ดับเบิลคลิกเพื่อ ติดตั้ง



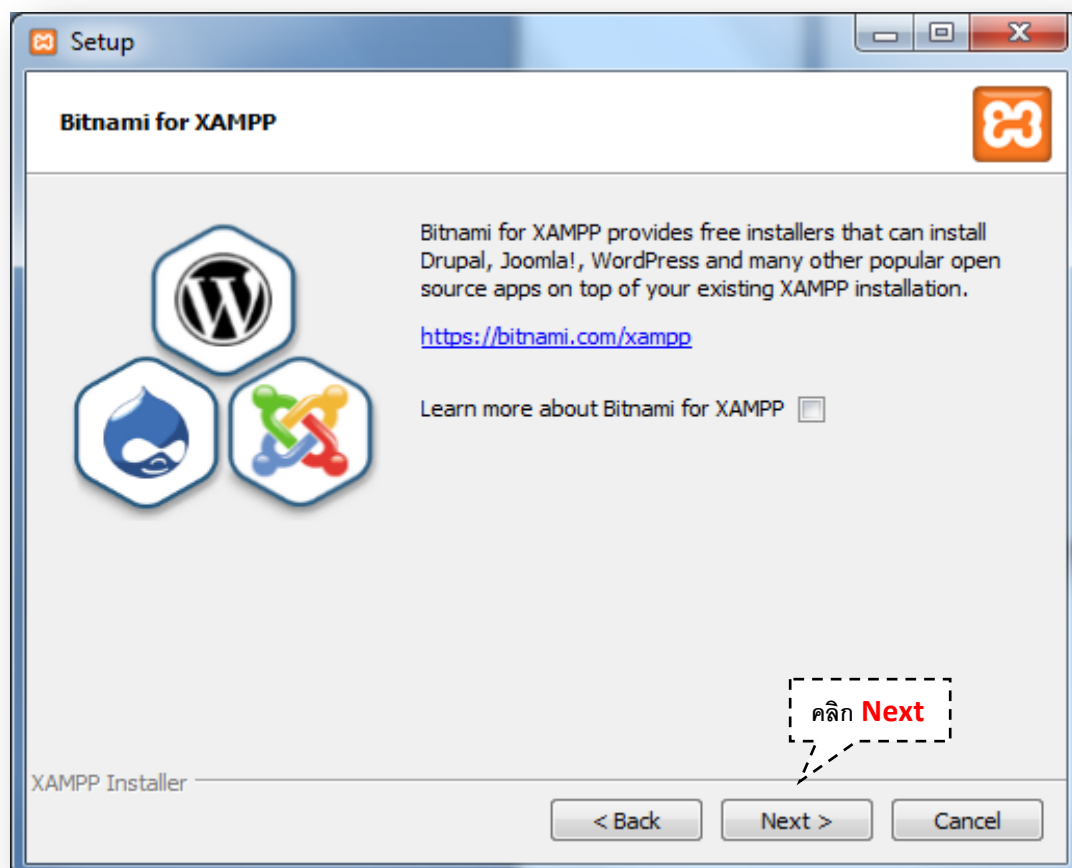
- จะเข้าสู่หน้าจอการติดตั้งโปรแกรม คลิก Next

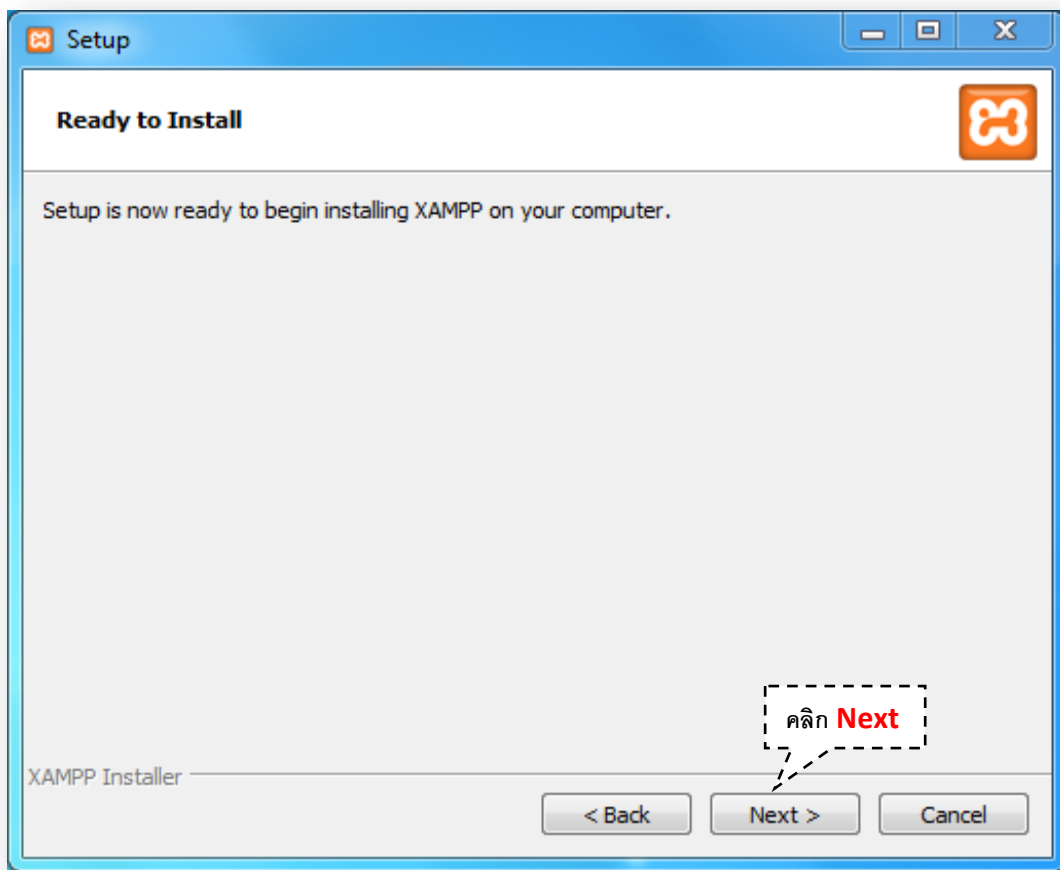


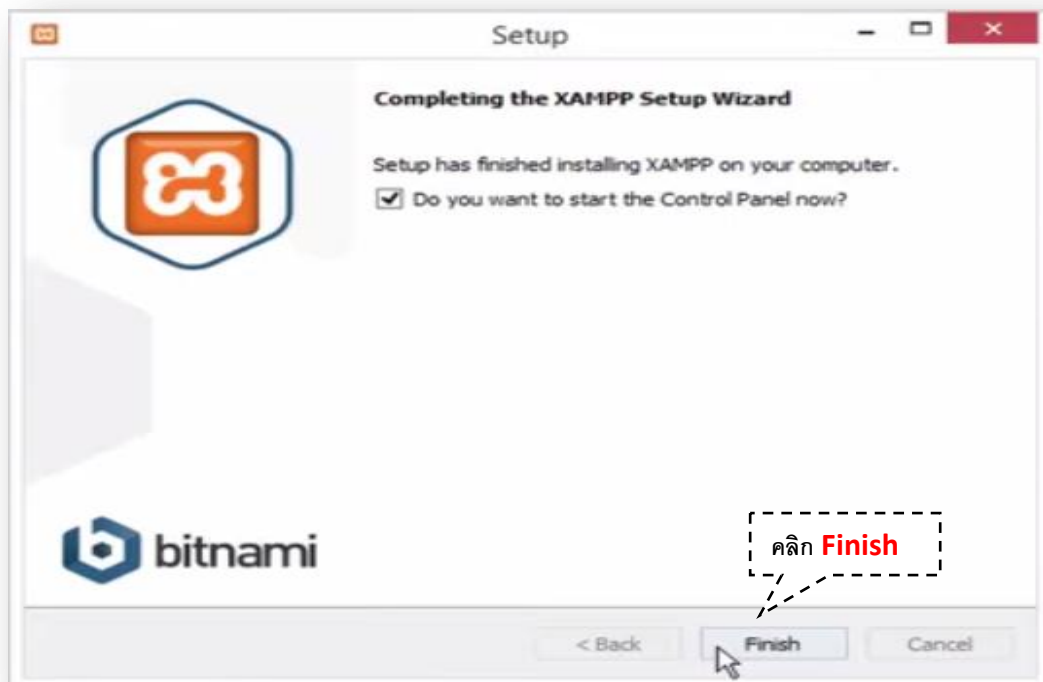
- เลือก Folder ที่อยู่ของโปรแกรม



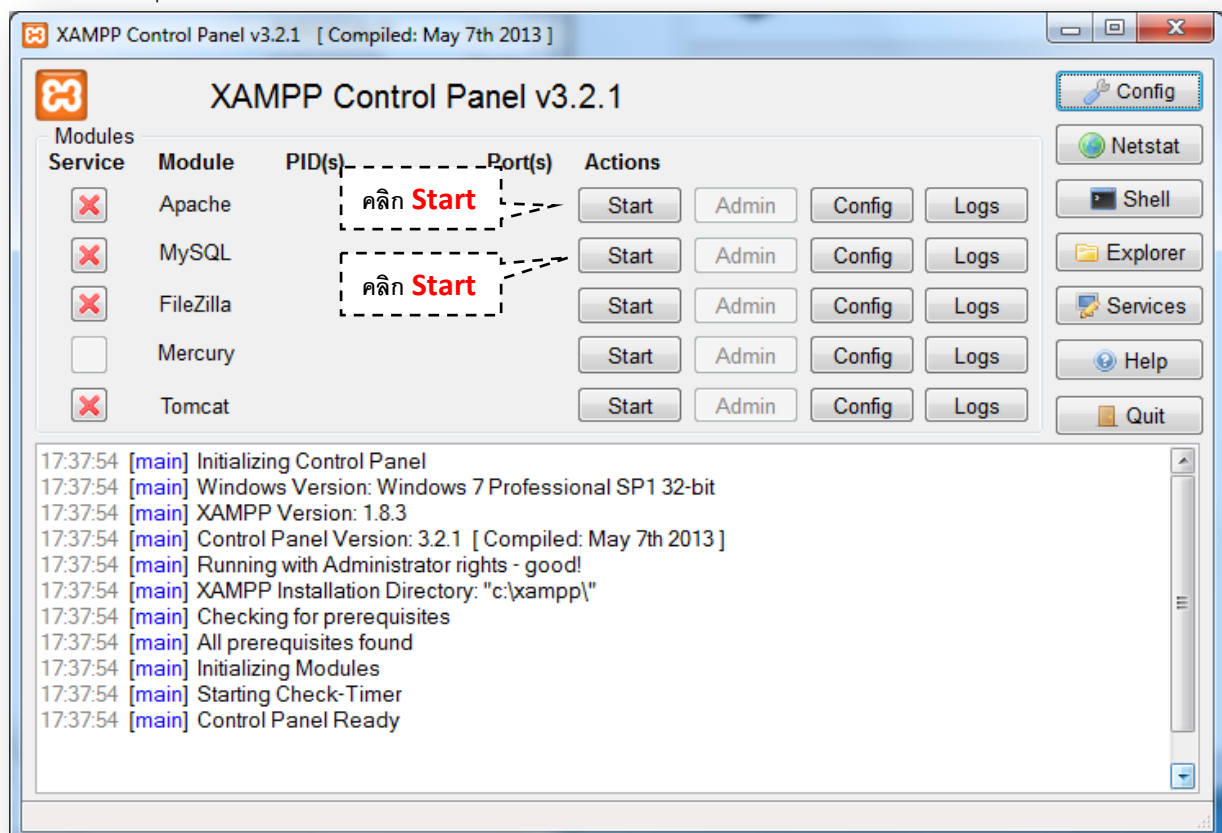
- คลิก next



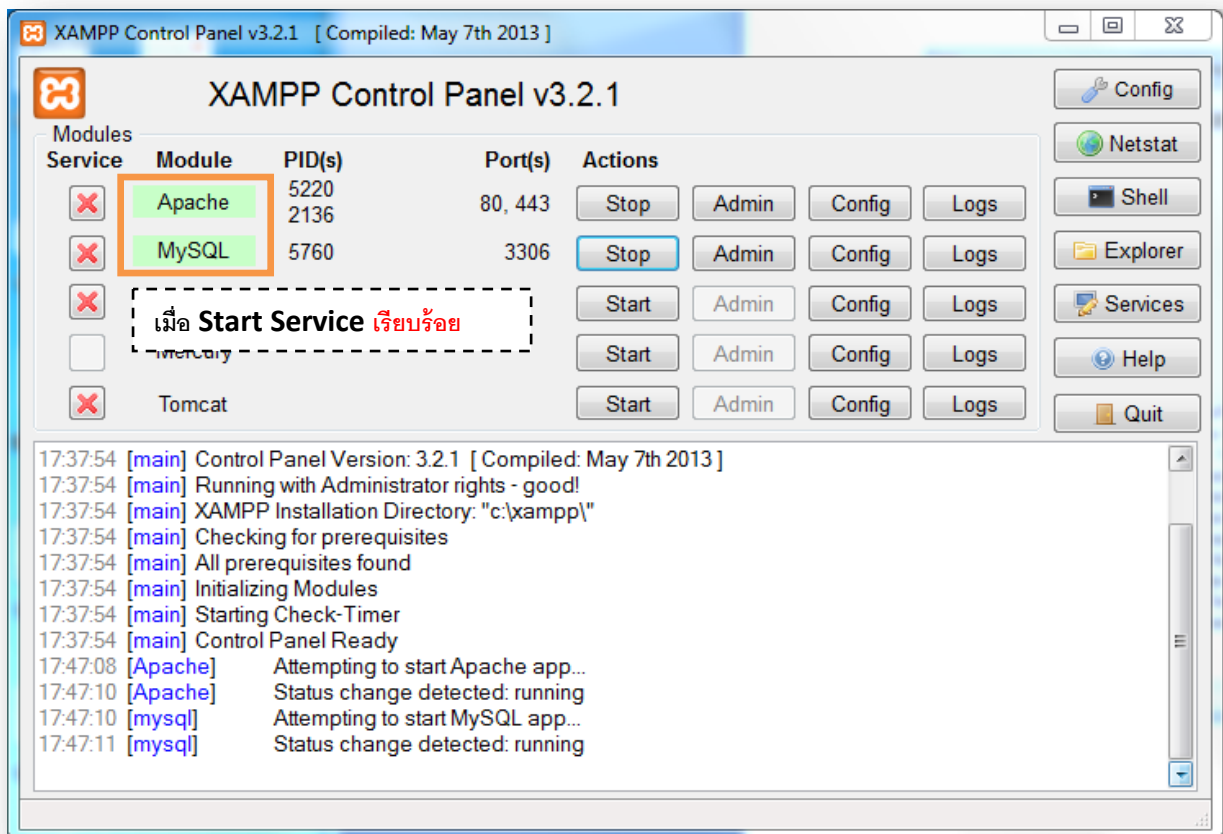




- คลิก Finish เพื่อเสร็จสิ้นการติดตั้งโปรแกรมจะแสดงหน้าจอ Xampp Control Panel ขึ้นมา
- คลิกปุ่ม Start Service Apache และ MySQL



- เมื่อ Start Service เรียบร้อยแล้ว



ทดสอบการใช้งาน เปิด Web Browser พิมพ์ localhost หรือ 127.0.0.1



เริ่มต้นการเขียนโปรแกรม

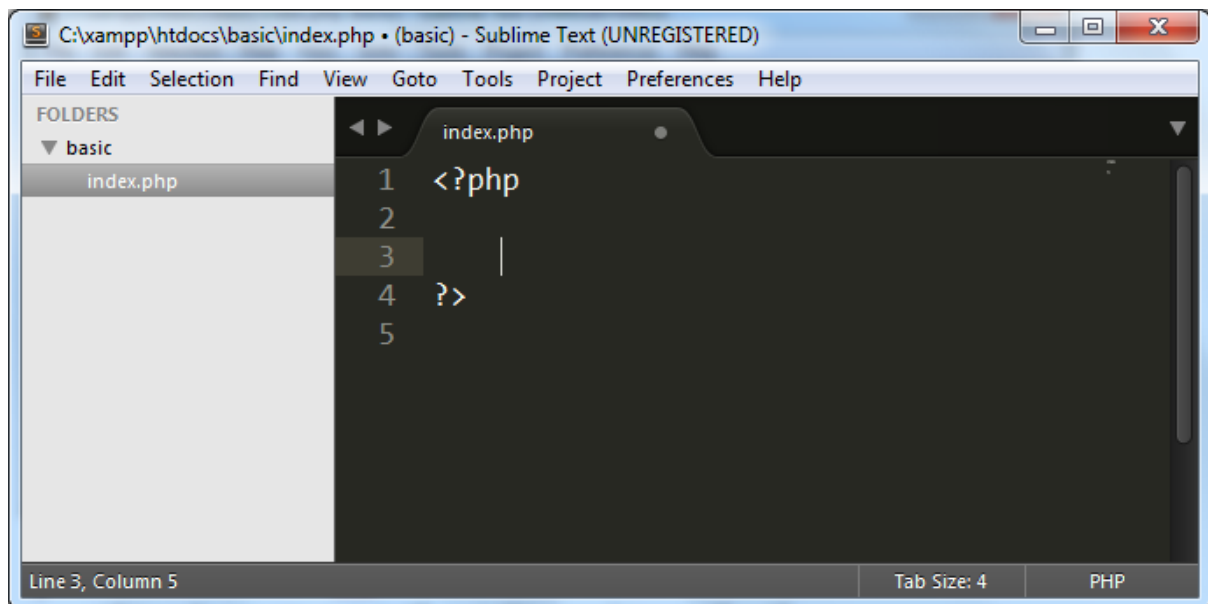
เมื่อติดตั้ง โปรแกรม Xampp และ โปรแกรม Sublime Text เรียบร้อยแล้ว
สร้าง Folder Project ของเราไว้ที่ C:xampp\htdocs/ชื่อ Folder Project ของเรา และสร้างไฟล์
index.php สำหรับทดสอบการทำงานของคำสั่งต่าง ๆ
ตัวอย่าง จะใช้ชื่อ Folder Project ว่า basic (C:xampp\htdocs/basic/index.php)

การเขียน Tag เปิด และ ปิดคำสั่ง PHP

ในการเขียนโปรแกรมภาษา PHP ก่อนเริ่มทำการเขียนต้องมีการเปิด Tag PHP กันก่อน
การเขียน Tag มี 2 รูปแบบด้วยกัน คือ <?php หรือ <? สำหรับการเปิด และ ?> สำหรับการปิด

ตัวอย่าง การเขียน Tag PHP

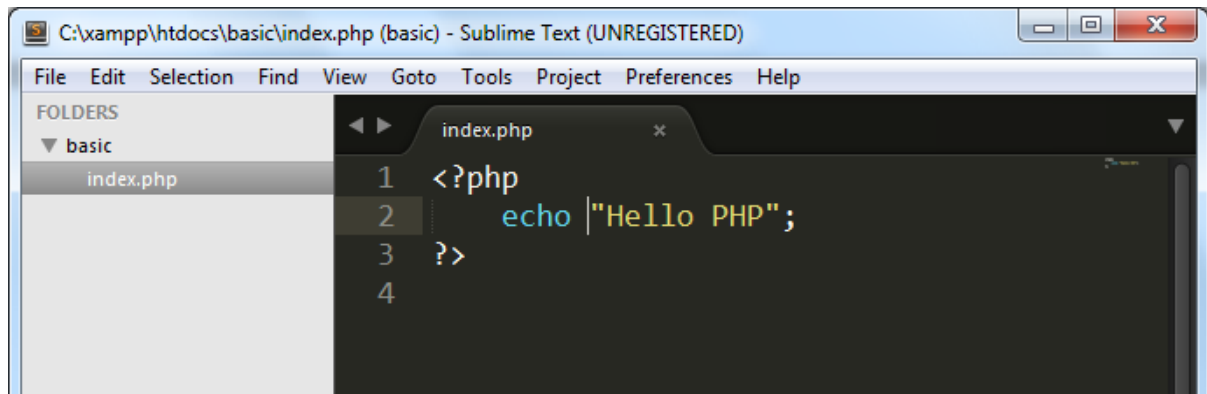
เราจะเขียนคำสั่ง PHP ภายในขอบเขตของ Tag <?php และเมื่อสิ้นสุดคำสั่ง จะจบด้วย ?> ทุก
ครั้ง เพื่อให้ตัวแปลภาษารู้ว่าส่วนใดเป็นคำสั่ง PHP นั่นเอง



ผู้เขียน แนะนำว่าให้เขียนแบบ <?php คำสั่ง ?> เนื่องจากการนำโปรแกรมไปใช้งานจริง บางครั้ง
เครื่อง Server ได้ทำการปิด Short Tag (<? ?>) ไว้ อาจทำให้ โปรแกรมไม่สามารถ ใช้งานได้

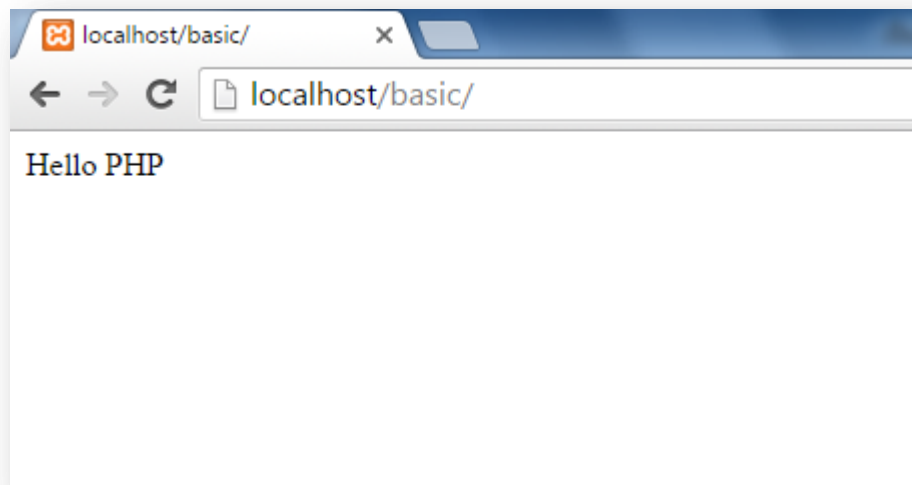
คำสั่ง `echo` คือคำสั่งสำหรับ การแสดงผลออกจากหน้าจอ

ตัวอย่าง 1.1 การแสดงผลข้อความด้วยคำสั่ง `echo`

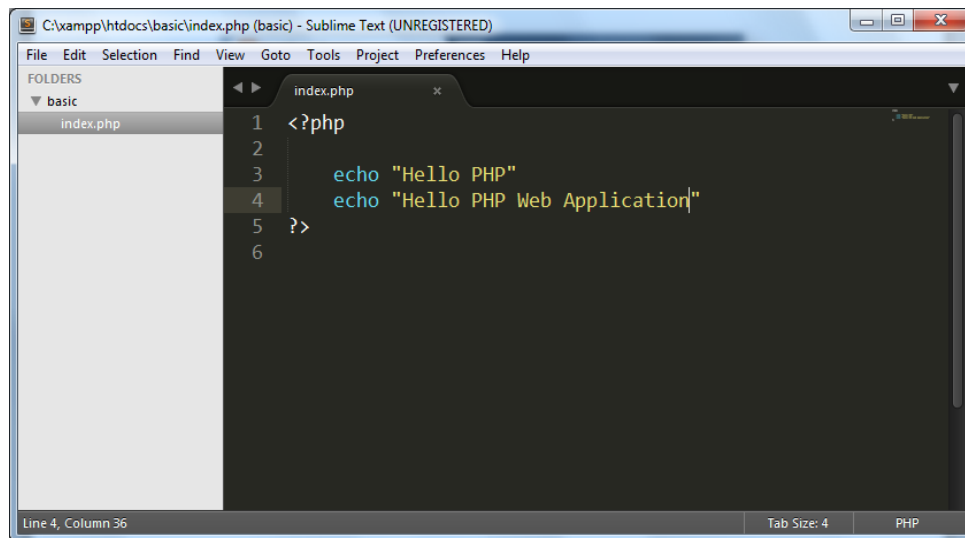


```
C:\xampp\htdocs\basic\index.php (basic) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
▼ basic
index.php
1 <?php
2     echo "Hello PHP";
3 ?>
4
```

ตัวอย่างการแสดงผลออกทางหน้าจอ



ในการจบประโยคคำสั่งโปรแกรมทุกครั้ง ต้องใส่เครื่องหมาย ; เซมิโคลอน ทุกครั้ง เพื่อเป็นการสิ้นสุดคำสั่ง



```
1 <?php
2
3     echo "Hello PHP"
4     echo "Hello PHP Web Application"
5 ?>
6
```

ตัวอย่าง Error กรณีลืมใส่ เซมิโคลอน ;



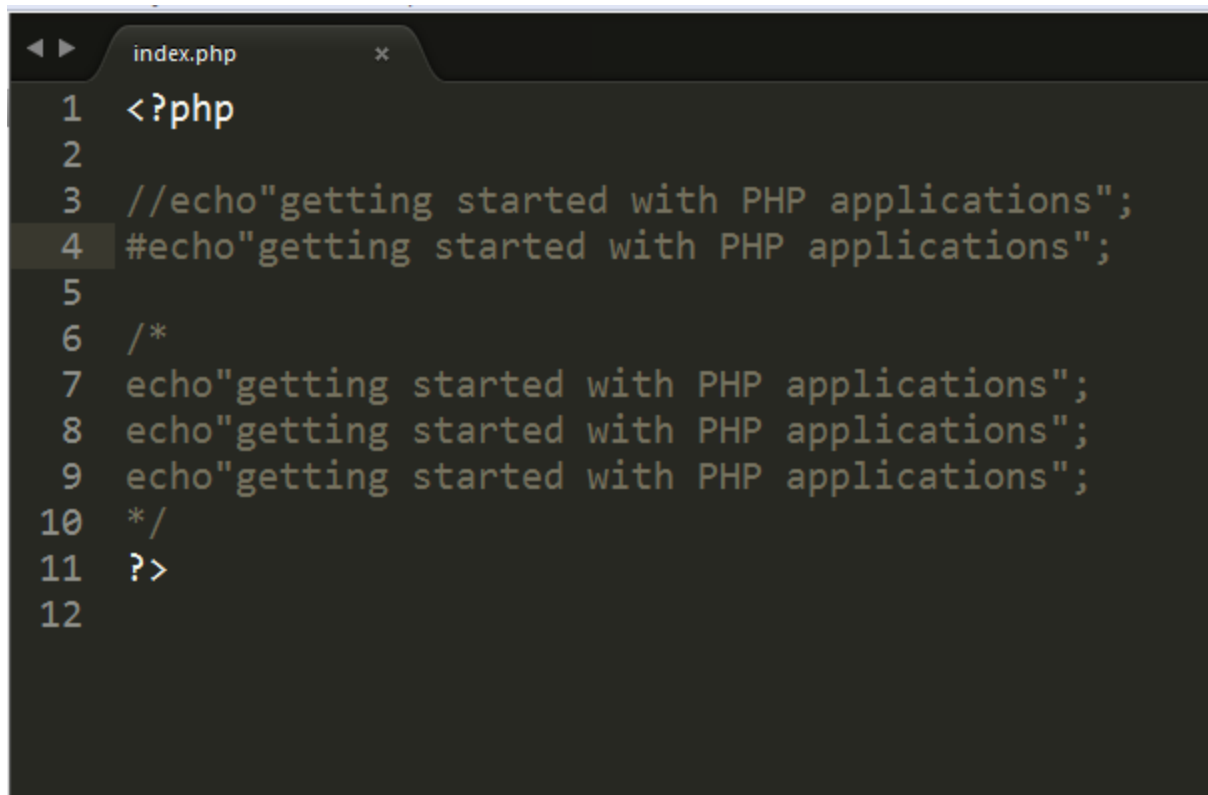
การเขียน Comment

การเขียน Comment คือการปิดการทำงานของคำสั่งบางอย่างที่เราไม่ต้องการให้ทำงาน หรือ เป็นการอธิบายการทำงานของคำสั่งโดยย่อ เพื่อความสะดวกในการกลับมาแก้ไขในภายหลัง

การเขียน comment ใน PHP มีรูปแบบดังนี้

1. การ Comment แบบ บรรทัดเดียว ใช้เครื่องหมาย // หรือ # คำสั่งที่อยู่ข้างหลังเครื่องหมายจะถูกปิดการทำงานทั้งบรรทัด
2. การ comment แบบ หลายบรรทัด ใช้เครื่องหมาย /* ตามด้วยคำสั่งที่ต้องการ ปิดการทำงาน และ ใช้เครื่องหมาย */ เพื่อจบการ comment

ตัวอย่างการ Comment

A screenshot of a code editor window titled 'index.php'. The code is written in PHP and demonstrates three different comment styles. Line 1 is the opening PHP tag. Line 2 is empty. Line 3 uses a double-slash comment to ignore an echo statement. Line 4 uses a hash comment to ignore another echo statement. Line 5 is empty. Line 6 starts a multi-line comment with '/*'. Lines 7, 8, and 9 contain echo statements that are commented out. Line 10 ends the multi-line comment with '*/'. Line 11 is the closing PHP tag, and line 12 is empty.

```
1 <?php
2
3 //echo"getting started with PHP applications";
4 #echo"getting started with PHP applications";
5
6 /*
7 echo"getting started with PHP applications";
8 echo"getting started with PHP applications";
9 echo"getting started with PHP applications";
10 */
11 ?>
12
```

การ เขียน PHP ร่วมกับ ภาษา HTML

คำสั่งของภาษา PHP นั้นสามารถ แทรกลงในคำสั่งที่เป็น Html ได้ หรือ สั่งให้แสดงผลเป็น Html ได้

ตัวอย่างการเขียน PHP ร่วมกับ HTML

```
index.php
1 <html>
2   <head>
3     <title>การเขียน PHP ร่วมกับ HTML</title>
4   </head>
5   <body>
6     <p><?php echo "start first my programing "; ?></p>
7   </body>
8 </html>
9
10
```

จะเห็นว่า Tag php <?php ?> เป็นตัวกำหนดขอบเขตของคำสั่งเพื่อให้ ตัวแปลภาษา รู้ว่าคำสั่งไหนคือ คำสั่ง PHP

```
index.php
1 <?php
2
3 echo"<ul>
4     <li>Red
5     <li>Green
6     <li>Blue
7     <li>Yellow
8   </ul>";
9 ?>
10
```

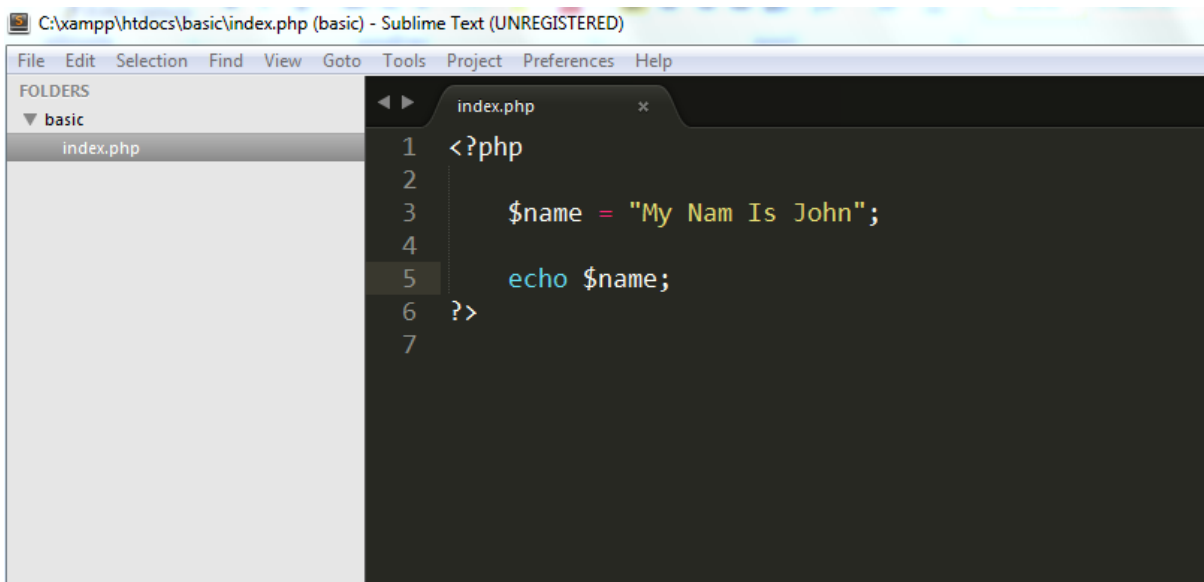
ตัวแปร (Variables)

ตัวแปร คือ ที่พักสำหรับเก็บข้อมูลเพื่อใช้ในการคำนวณ หรือ แสดงผลข้อมูล ภายในระบบ โดยในภาษา PHP การสร้างตัวแปรจะมีเครื่องหมาย \$ นำหน้า และตามด้วยชื่อตัวแปร เช่น \$name; การกำหนดค่าให้กับตัวแปร เราสามารถ กำหนดค่าเริ่มต้นให้กับตัวแปรได้ เช่น \$name = "Hello PHP"; ตัวแปร name ก็จะเก็บค่าข้อความ

ข้อแนะนำการตั้งชื่อตัวแปร

- ชื่อของตัวแปรนั้นจะต้องขึ้นต้นด้วยเครื่องหมาย \$ (Dollar Sign) และตามด้วยตัวอักษร ไม่เป็นตัวเลขหรือสัญลักษณ์พิเศษ
- ชื่อของตัวแปรควรจะสื่อความหมาย กับการไปใช้งาน เช่น \$fullname , \$phone
- ชื่อของตัวแปรชื่อเดียวกันแต่ใช้ตัวอักษรพิมพ์เล็กพิมพ์ใหญ่ถือว่าเป็นคนละตัวแปรกันเช่น \$price , \$PRICE

ตัวอย่างการสร้างตัวแปร

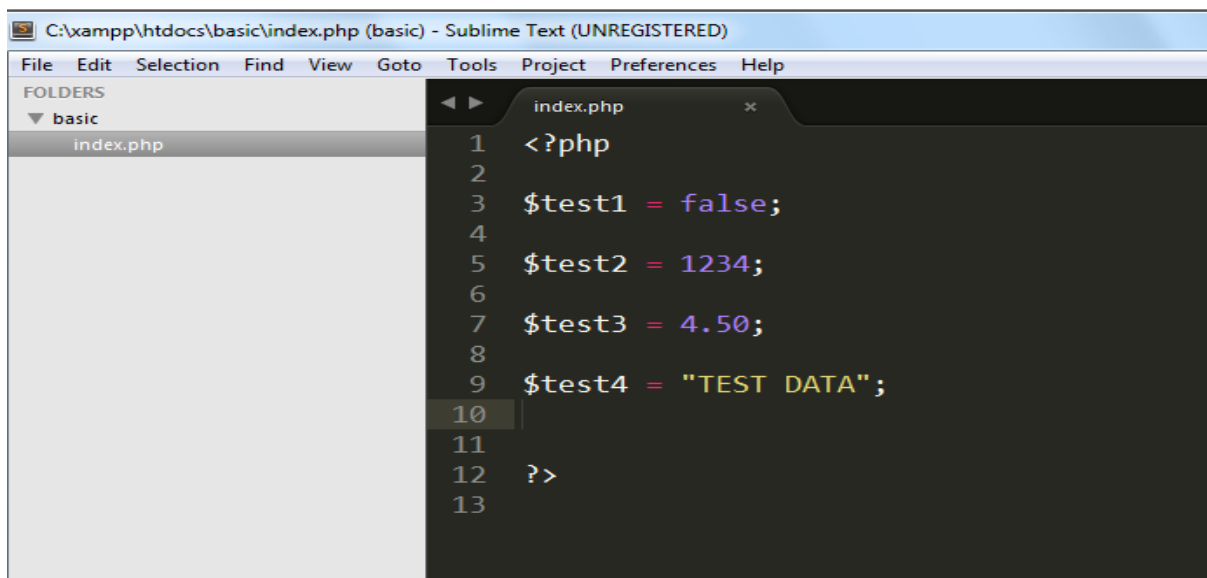


```
C:\xampp\htdocs\basic\index.php (basic) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
▼ basic
index.php
1 <?php
2
3     $name = "My Nam Is John";
4
5     echo $name;
6 ?>
7
```

ชนิดของข้อมูลในตัวแปร

- Boolean เก็บข้อมูลเป็นค่า จริง true หรือ เท็จ false
- Integer เก็บข้อมูลเป็นตัวเลขจำนวนเต็ม
- Float , double เก็บข้อมูลเป็นเลขจำนวนจริง
- String เก็บข้อมูลที่เป็นข้อความ
- Array เก็บข้อมูลหลาย ๆ ค่าไว้ในตัวแปรเดียว

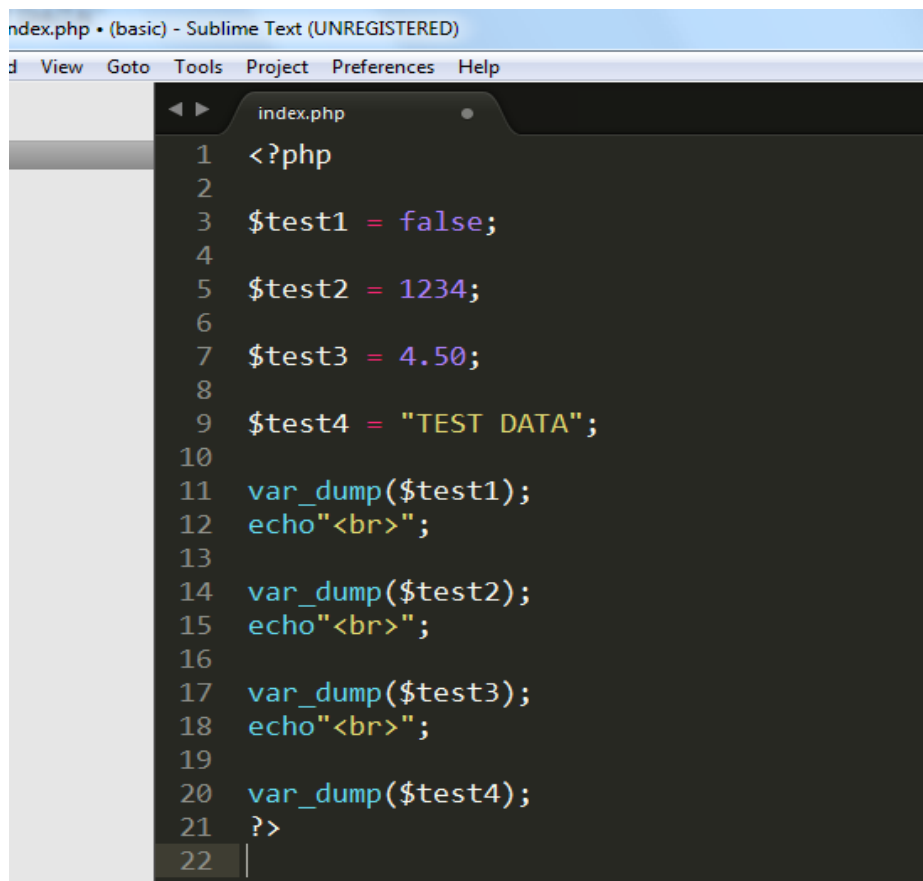
ตัวอย่างการเก็บข้อมูลรูปแบบต่าง ๆ ในตัวแปร



```
C:\xampp\htdocs\basic\index.php (basic) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  basic
    index.php
index.php
1 <?php
2
3 $test1 = false;
4
5 $test2 = 1234;
6
7 $test3 = 4.50;
8
9 $test4 = "TEST DATA";
10
11
12 ?>
13
```

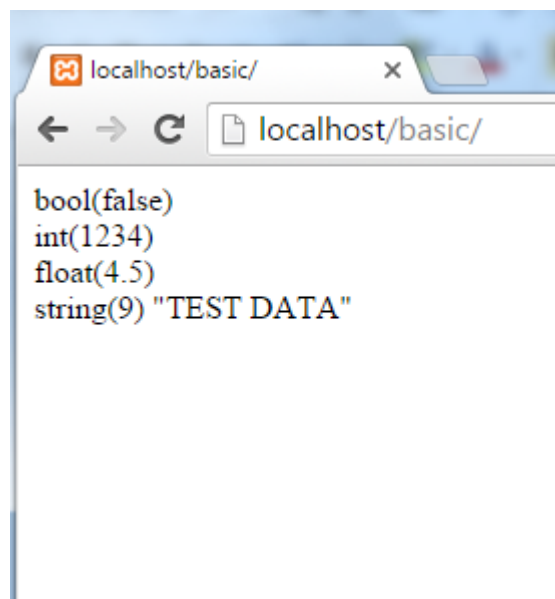
ในการประกาศตัวแปรบางภาษาโปรแกรมคอมพิวเตอร์อาจจำเป็นต้องระบุให้ชัดเจนว่าตัวแปรที่เราได้สร้างขึ้นมาเป็นตัวแปรชนิดใด แต่สำหรับในภาษา PHP นั้นไม่จำเป็นต้องระบุชนิดของตัวแปร ระบบสามารถทราบได้เองว่าเป็นตัวแปรชนิดใด ซึ่งในการเขียนโปรแกรมนั้นเราจำเป็นต้องทราบว่าตัวแปรที่ใช้งานอยู่นั้นเก็บข้อมูลชนิดใดอยู่สามารถตรวจสอบได้โดยใช้คำสั่ง `var_dump` เพื่อตรวจสอบว่าตัวแปรที่ใช้งานเก็บข้อมูลแบบใด

ตัวอย่างการใช้งานคำสั่ง var_dump เพื่อตรวจสอบชนิดของตัวแปร



```
1 <?php
2
3 $test1 = false;
4
5 $test2 = 1234;
6
7 $test3 = 4.50;
8
9 $test4 = "TEST DATA";
10
11 var_dump($test1);
12 echo"<br>";
13
14 var_dump($test2);
15 echo"<br>";
16
17 var_dump($test3);
18 echo"<br>";
19
20 var_dump($test4);
21 ?>
22
```

ตัวอย่างการแสดงผลเมื่อทำงานบน Web Browser



ตัวดำเนินการ (Operators)

ตัวดำเนินการ คือ เครื่องหมายที่ใช้ในการจัดการกับข้อมูลภายในตัวแปร เพื่อ กำหนดค่า คำนวณ เปรียบเทียบ เพิ่มค่า ลดค่า และ อื่น ๆ ตัวอย่างตัวดำเนินการเช่น เครื่องหมาย บวก ลบ คูณ หาร มากกว่า น้อยกว่า เป็นต้น

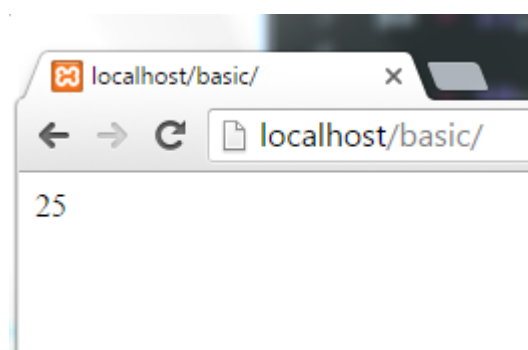
ตัวอย่างตัวดำเนินการทางคณิตศาสตร์

เราสามารถใช้งานเครื่องหมายทางคณิตศาสตร์เพื่อคำนวณค่าของตัวแปรได้ โดยกำหนดให้ตัวแปร \$a มีค่า เท่ากับ 15 และ ตัวแปร \$b = 10

```
index.php
1 <?php
2
3 $a = 15;
4
5 $b = 10;
6
7 echo $c = $a + $b;
8
9 echo $c = $a - $b;
10
11 ?>
12
```

ในตัวอย่าง นำค่าตัวแปร \$a มาบวก กับ \$b และนำค่าที่ได้ มากำหนดให้กับตัวแปร \$c ผลที่ได้คือ 15 + 10 ดังนั้น ตัวแปร \$c เก็บค่า 25

ผลการทำงาน



ตัวดำเนินการทางคณิตศาสตร์มีดังนี้

- กำหนดให้ตัวแปร \$a\$ มีค่าเท่ากับ 10 ตัวแปร \$b\$ มีค่าเท่ากับ 5

ตัวดำเนินการ	ตัวอย่างการใช้งาน	ผลที่ได้
+ (บวก)	$\$a + \b	25
- (ลบ)	$\$a - \b	5
* (คูณ)	$\$a * \b	150
/ (หาร)	$\$a / \b	1.5
% (หารเอาแต่เศษ)	$\$a \% \b	5

ตัวดำเนินการเปรียบเทียบ

- เป็นการเปรียบเทียบค่าในตัวแปรหากเป็นจริงจะให้ค่า จริง (true)
หรือไม่เป็นจริงจะได้ค่า เท็จ (false)
- ตัวอย่าง $\$a == \b ไม่เป็นความจริง เนื่องจาก ค่า 15 ไม่เท่ากับ 5 จึงได้ค่าเท็จ (false)

ตัวดำเนินการ	ตัวอย่างการใช้งาน	ผลที่ได้
== (เท่ากับ)	$\$a == \b	เท็จ(false)
=== (เท่ากับ) (ชนิดข้อมูลต้องเป็นเดียวกัน)	$\$a === \b	เท็จ(false)
!= (ไม่เท่ากับ)	$\$a != \b	จริง(true)
> (มากกว่า)	$\$a > \b	จริง(true)
< (น้อยกว่า)	$\$a < \b	เท็จ(false)
>= (มากกว่าหรือเท่ากับ)	$\$a >= \b	จริง(true)
<= (น้อยกว่าหรือเท่ากับ)	$\$a <= \b	เท็จ(false)

ตัวดำเนินการตรรกศาสตร์

ตัวดำเนินการ	ตัวอย่างการใช้งาน	ผลที่ได้
(หรือ)	$\$a == \b	เท็จ(false)
or (หรือ)	$\$a === \b	เท็จ(false)
&& (และ)	$\$a != \b	จริง(true)
and (และ)	$\$a > \b	จริง(true)
! (ไม่)	$\$a < \b	เท็จ(false)

ตัวดำเนินการ เพิ่มค่า ลด ค่า

ตัวดำเนินการ	ตัวอย่างการใช้งาน	ผลที่ได้
+=	$\$a += \b	$\$a = \$a + \$b$
-=	$\$a -= \b	$\$a = \$a - \$b$
*=	$\$a *= \b	$\$a = \$a * \$b$
/=	$\$a /= \b	$\$a = \$a / \$b$
%=	$\$a \% = \b	$\$a = \$a \% \$b$

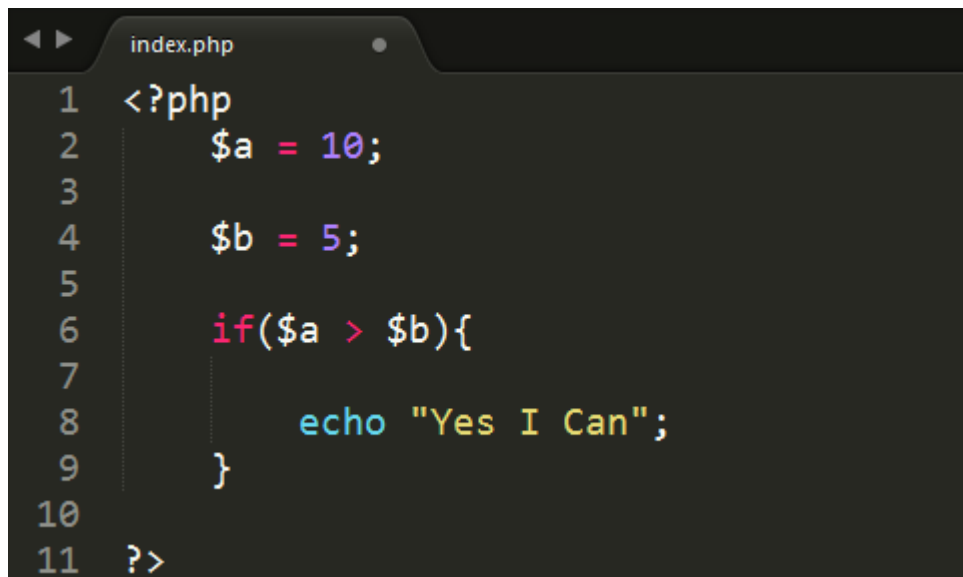
คำสั่งตรวจสอบเงื่อนไข if else

คำสั่งตรวจสอบเงื่อนไข คือ การตรวจสอบข้อมูลเพื่อเปรียบเทียบค่าที่ต้องการแสดงผล ว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงให้ทำแบบนี้ หรือ ไม่เป็นจริงให้ทำอีกแบบหนึ่ง เช่น หากวันนี้เป็นวันหยุดฉันจะไม่ไปโรงเรียน หากเป็นจริง ฉันจะไม่ไปโรงเรียน หากไม่เป็นจริง ฉันจะไปโรงเรียน หรือ ถ้าเด็กชาย มีอายุมากกว่า 15 ปี ถ้าเป็นจริงให้ใช้คำนำหน้าว่า นาย ถ้าไม่เป็นจริง ให้ใช้คำนำหน้าว่า เด็กชาย เป็นต้น

ในการเขียนโปรแกรมนั้นเราจะทำการตรวจสอบค่าของตัวแปร เพื่อให้แสดงผลข้อมูลตามเงื่อนไขต่าง ๆ

คำสั่ง if

- คือ คำสั่งตรวจสอบเงื่อนไข และทำงานเมื่อเงื่อนไขเป็นจริงเท่านั้น



```
index.php
1  <?php
2      $a = 10;
3
4      $b = 5;
5
6      if($a > $b){
7
8          echo "Yes I Can";
9      }
10
11  ?>
```

จากตัวอย่าง เราได้กำหนดค่าตัวแปร \$a มีค่าเท่ากับ 10 และ \$b มีค่าเท่ากับ 5

บรรทัดที่ 2 กำหนดค่าตัวแปร \$a มีค่าเท่ากับ 10

บรรทัดที่ 4 กำหนดค่าตัวแปร \$b มีค่าเท่ากับ 5

บรรทัดที่ 6 คำสั่ง if กำหนดเงื่อนไข \$a มีค่า มากกว่า \$b จริงหรือไม่ หากเป็นจริงจะทำคำสั่งในวงเล็บ { }

บรรทัดที่ 8 หาก \$a มีค่ามากกว่า \$b จริง จะทำคำสั่ง echo "Yes I Can" หากไม่จริงจะข้ามการทำงานส่วนนี้ไป

คำสั่ง if else

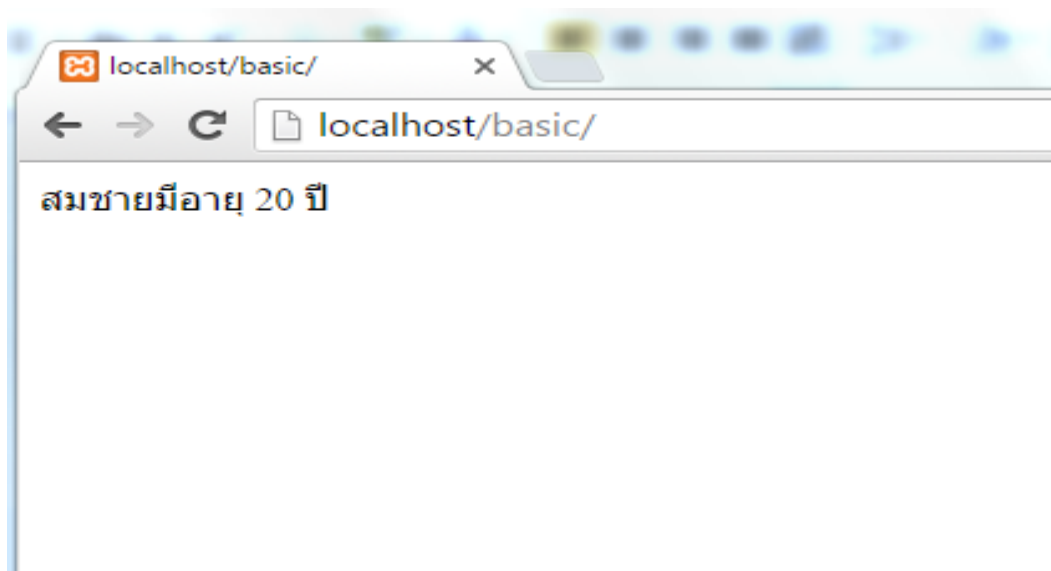
คำสั่งตรวจสอบเงื่อนไขโดยจะทำคำสั่งที่อยู่ภายใน if เมื่อเงื่อนไขเป็นจริง และทำคำสั่งที่อยู่ภายใน else เมื่อเงื่อนไขเป็นเท็จ

ตัวอย่างการใช้งาน คำสั่ง if else

กำหนดให้ตัวแปร \$old มีค่าเท่ากับ 20 คำสั่ง if ตรวจสอบเงื่อนไข ถ้า ตัวแปร \$old มีค่ามากกว่าหรือเท่ากับ 15 เป็นจริงให้ทำคำสั่งที่อยู่ภายใน if หากไม่เป็นจริงให้ทำคำสั่ง ที่อยู่ภายใน else

```
index.php x
1 <?php
2
3 $old = 20;
4
5 if($old >= 15 ){
6
7     echo"สมชายมีอายุ " . $old . " ปี";
8
9 }else{
10
11     echo "สมชายมีอายุต่ำกว่า 15 ปี";
12
13 }
14 ?>
```

ผลการทำงาน



คำสั่ง if ซ้อน if

คำสั่งตรวจสอบเงื่อนไข แบบ ซ้อนซ้อน คำสั่ง if สามารถตรวจสอบเงื่อนไขได้หลายชั้น

ตัวอย่างการใช้งาน if else if

ตัวอย่างการแบ่งเกรด โดยในคำสั่ง if จะทำการตรวจสอบเงื่อนไข ถ้าจริงจะทำงานภายในคำสั่ง
ปีกกาของ if ถ้าไม่จริง จะตรวจสอบเงื่อนไขของ else if ไปเรื่อย ๆ ถ้าไม่ตรงกับเงื่อนไขใดเลยจะทำคำสั่งใน
ปีกกา ของ else

```
index.php
1 <?php
2
3 $score = 10;
4
5 if($score >= 85){
6
7     echo "คุณทำข้อสอบได้ ".$score." คะแนน คุณได้ Grade 4";
8
9 }else if($score >=70){
10
11     echo "คุณทำข้อสอบได้ ".$score." คะแนน คุณได้ Grade 3";
12
13 }else if($score >= 60){
14
15     echo "คุณทำข้อสอบได้ ".$score." คะแนน คุณได้ Grade 2";
16
17 }else if($score >= 50){
18
19     echo "คุณทำข้อสอบได้ ".$score." คะแนน คุณได้ Grade 1";
20
21 }else{
22     echo "คุณทำข้อสอบได้ ".$score." คะแนน คุณไม่ผ่านการทดสอบ";
23 }
24
25 ?>
```

บรรทัดที่ 3 กำหนดค่า \$score มีค่าเท่ากับ 10

บรรทัดที่ 5 ตรวจสอบเงื่อนไข ถ้า \$score มีค่า มากกว่าหรือ เท่ากับ 85 ให้ทำคำสั่งภายในเงื่อนไขนี้

บรรทัดที่ 9 ตรวจสอบเงื่อนไข ถ้า \$score มีค่า มากกว่าหรือ เท่ากับ 70 ให้ทำคำสั่งภายในเงื่อนไขนี้

บรรทัดที่ 13 ตรวจสอบเงื่อนไข ถ้า \$score มีค่า มากกว่าหรือ เท่ากับ 60 ให้ทำคำสั่งภายในเงื่อนไขนี้

บรรทัดที่ 17 ตรวจสอบเงื่อนไข ถ้า \$score มีค่า มากกว่าหรือ เท่ากับ 50 ให้ทำคำสั่งภายในเงื่อนไขนี้

บรรทัดที่ 21 หากไม่ตรงกับเงื่อนไขใดเลย ให้ทำคำสั่ง ภายใน else

คำสั่ง switch case

คำสั่ง เพื่อใช้งานการเลือกทำงาน 1 ทางเลือกจากหลาย ๆ ทางเลือกโดยจะตรวจสอบว่าตรงกับ case ไດ ก็ทำงานในคำสั่งนั้น เราจะใช้คำสั่ง break เพื่อหยุดการทำงานของ case และ คำสั่ง default เมื่อ เงื่อนไขไม่ตรงกับ case ไດ ๆ

ตัวอย่างการใช้งาน

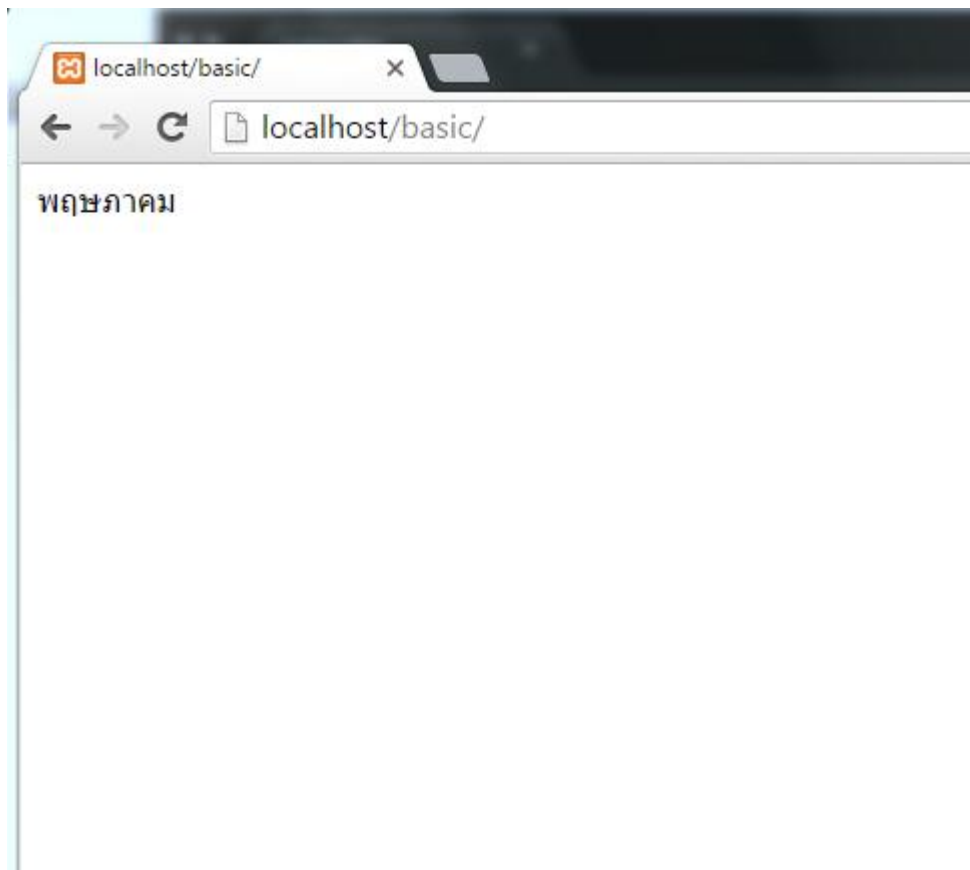
ในการใช้งานคำสั่ง switch เราจำเป็นต้องระบุ คำสั่ง break หลังจากคำสั่ง case เพื่อหยุดการทำงาน หากไม่ระบุ คำสั่ง break จะแสดง case ต่อไป หลังจาก case ที่ ตรงกับเงื่อนไข และ ระบุคำสั่ง default เมื่อ ไม่ตรงกับเงื่อนไขใด ให้ทำงานในคำสั่ง default

```
index.php
1  <?php
2
3  $month = 5;
4
5  switch ($month) {
6      case 1: echo "มกราคม"; break;
7      case 2: echo "กุมภาพันธ์"; break;
8      case 3: echo "มีนาคม"; break;
9      case 4: echo "เมษายน"; break;
10     case 5: echo "พฤษภาคม"; break;
11     case 6: echo "มิถุนายน"; break;
12     case 7: echo "กรกฎาคม"; break;
13     case 8: echo "สิงหาคม"; break;
14     case 9: echo "กันยายน"; break;
15     case 10: echo "ตุลาคม"; break;
16     case 11: echo "พฤศจิกายน"; break;
17     case 12: echo "ธันวาคม"; break;
18
19     default:
20         # code...
21         echo "ข้อมูลเดือนไม่ถูกต้อง";
22     break;
23 }
24
25 ?>
```

จากตัวอย่าง กำหนดค่าตัวแปร \$month มีค่าเท่ากับ 5 ตรวจสอบเงื่อนไขภายใน คำสั่ง switch หากตรงกับ case ไດ ๆ ให้แสดงคำสั่งภายใน case นั้น

ตัวอย่างการทำงาน

ในตัวอย่าง ตัวแปร \$month มีค่าเท่ากับ 5 เมื่อทำคำสั่ง switch จะตรงกับ case ที่มีเงื่อนไขตรงกับ 5 และหยุดการทำงาน ด้วยคำสั่ง break



ความแตกต่างระหว่าง คำสั่ง switch กับ คำสั่ง if else if

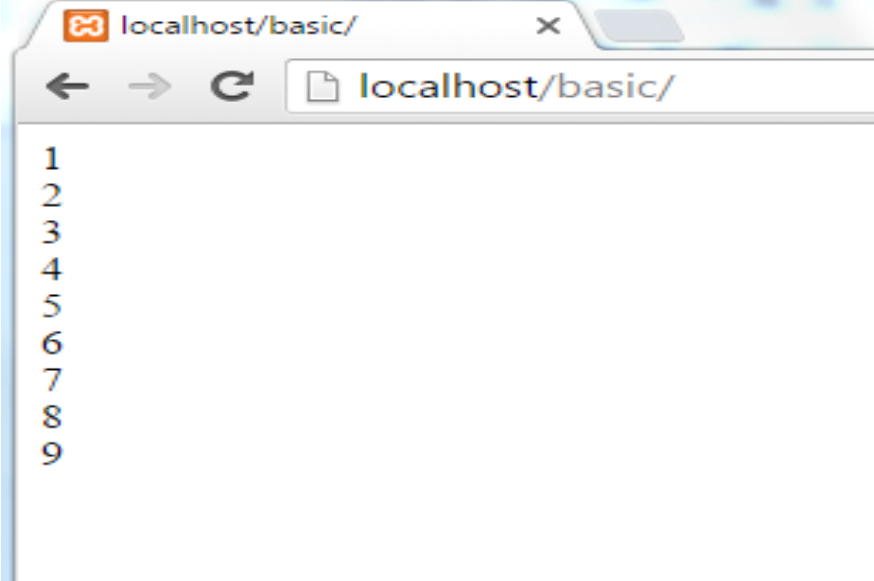
คำสั่ง switch สามารถตรวจสอบเงื่อนไข ของแต่ละ case โดยที่ไม่มีความเกี่ยวข้องกัน คำสั่ง if else if จำเป็นต้องเปรียบเทียบ ค่า ว่าเป็นจริงหรือเท็จ ก่อนที่จะทำคำสั่งต่อไป

คำสั่งทำซ้ำ for

คำสั่ง for จะมีการวนรอบการทำงาน จนกว่าจะครบเงื่อนไขตามที่กำหนดถึงจะหยุดการทำงาน
ตัวอย่างคำสั่ง for แสดงตัวเลข 1 ถึง 10

```
index.php
1  <?php
2
3      for($i = 1; $i<10; $i++){
4
5          echo $i;
6          echo"<br>";
7      }
8
9  ?>
10
```

ตัวอย่างการทำงาน



localhost/basic/

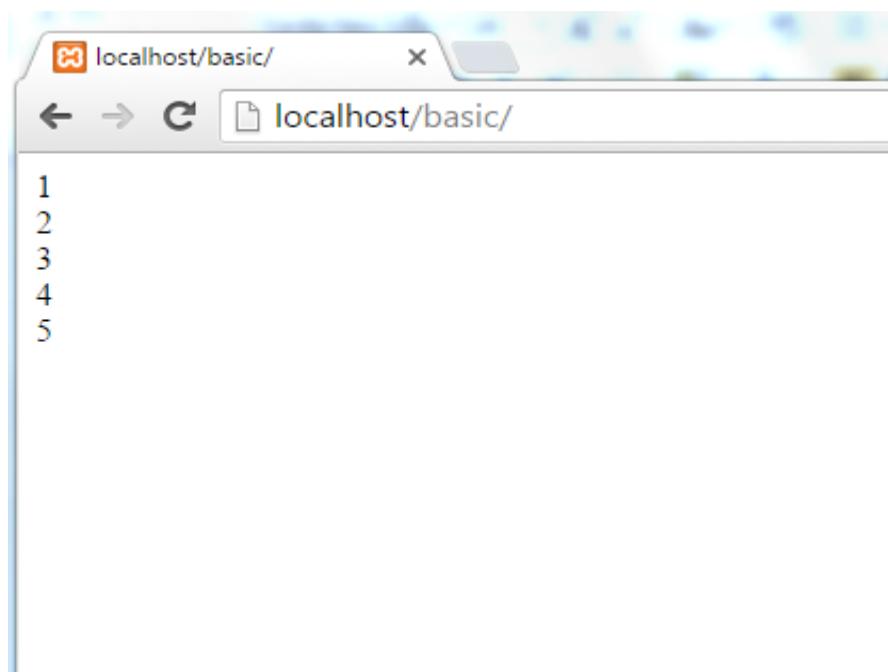
localhost/basic/

1
2
3
4
5
6
7
8
9

เราสามารถใส่คำสั่ง if เพื่อตรวจสอบเงื่อนไข และ ใช้คำสั่ง break เพื่อหยุดการทำงานของ for ตัวอย่าง ต้องการ แสดงตัวเลข 1 ถึง 5

```
index.php
1 <?php
2
3     for($i = 1; $i<10; $i++){
4
5         echo $i;
6         echo"<br>";
7
8         if($i == 5){
9             break;
10        }
11    }
12
13 ?>
14
```

ตัวอย่างการทำงาน



localhost/basic/

localhost/basic/

1
2
3
4
5

คำสั่ง while

คือคำสั่งตรวจสอบเงื่อนไข และจะวนรอบการทำงานจนกว่าเงื่อนไขจะเป็นเท็จ
ตัวอย่างการใช้งานคำสั่ง while



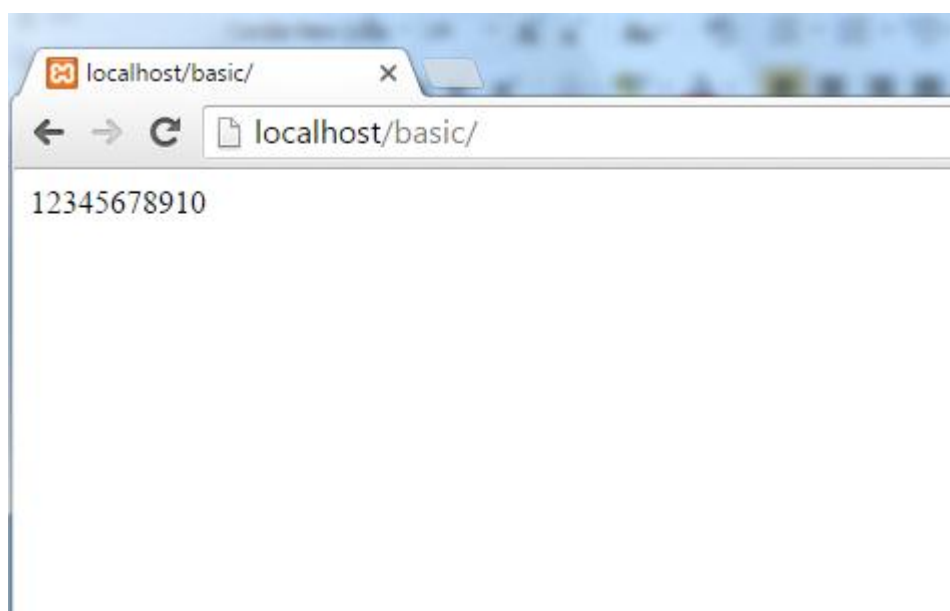
```
1 <?php
2
3     $i = 1;
4
5     while($i<=10){
6
7         echo $i;
8
9         $i++;
10    }
11 ?>
```

บรรทัดที่ 3 กำหนดค่าเริ่มต้น ตัวแปร \$i มีค่าเท่ากับ 1

บรรทัดที่ 5 คำสั่ง while ตรวจสอบค่า \$i มี ค่าน้อยกว่า 10 ให้ทำคำสั่งด้านล่าง หรือ เท่ากับ 10 ให้หยุดการทำงาน

บรรทัดที่ 7 คำสั่ง echo แสดงค่าตัวแปร \$i

บรรทัดที่ 9 ใช้ตัวดำเนินการเพิ่มค่าตัวแปร \$i ทีละ 1



คำสั่ง do while

เป็นคำสั่งวนรอบการทำงาน แต่จะทำ คำสั่ง do ก่อน หนึ่งครั้ง แล้วค่อยตรวจสอบเงื่อนไข ถ้าเป็นจริงจะทำต่อจนกว่าเงื่อนไขจะเป็นเท็จ และหยุดการทำงาน

```
index.php x
1 <?php
2
3 $i = 1;
4
5 $sum = 0;
6
7     do{
8         echo $i."</br>";
9
10        $sum += $i;
11
12        $i++;
13
14    }while($i <= 5);
15
16    echo "ผลรวมเท่ากับ ".$sum;
17 ?>
18
```

ตัวแปรชนิด Array

เป็นตัวแปรที่สามารถเก็บข้อมูล หลาย ๆ ตัวไว้ในตัวแปรเดียว เปรียบเทียบได้กับลิ้นชัก มีช่องอยู่หลายช่อง ในลิ้นชัก สามารถ เก็บของได้หลายช่อง แต่กับตัวแปร Array สามารถ เก็บกี่ช่องก็ได้ และจะต้องมีชื่อของแต่ละช่องเพื่อให้สามารถค้นหาสิ่งของที่อยู่ภายในได้ถูกต้อง

การใช้งานตัวแปร Array

ตัวแปร Array จะประกอบด้วย Key และ Value

Key คือตำแหน่งของข้อมูลใน Array หากเราไม่ได้กำหนด Key ตัวแรกของ Array จะเริ่มต้นที่ 0

Value คือ ค่าที่เก็บอยู่ในตัวแปร

```
$book = ['NodeJs', 'Java', 'PHP', 'jQuery'];
```

ตัวอย่างข้อมูลที่เก็บอยู่ในตัวแปร \$book

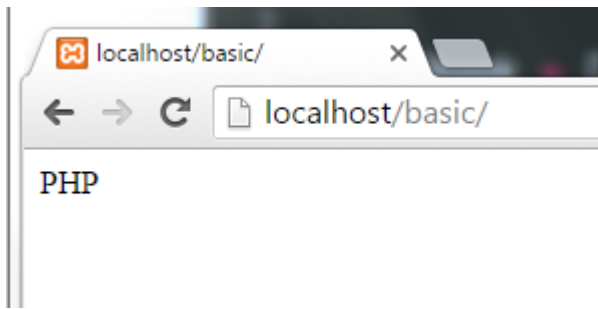
Key	Value
0	NodeJs
1	Java
2	PHP
3	jQuery

เราสามารถเข้าถึงข้อมูลภายใน Array ที่ละตัวได้โดยกำหนด Key ที่ต้องการ

ตัวอย่าง \$book[2] จะได้ผลลัพธ์คือ ข้อมูลที่อยู่ลำดับที่ 2 ใน Array โดยเริ่มนับจาก 0

```
1 <?php
2
3     $book = ['NodeJs', 'Java', 'PHP', 'jQuery'];
4
5     echo $book['2'];
6
7 ?>
```

การแสดงผลข้อมูลที่อยู่ในตัวแปร \$book ในลำดับที่ 2 คือ PHP



การกำหนด Key และ Value ให้กับ ตัวแปร Array ตัวอย่างกำหนด Key คือชื่อหนังสือ และ Value คือจำนวนหนังสือ

```
index.php
1 <?php
2
3     $book = [
4         'NodeJs'=>'5',
5         'Java'=>'3',
6         'PHP'=>'10',
7         'jQuery'=>'6'
8     ];
9
```

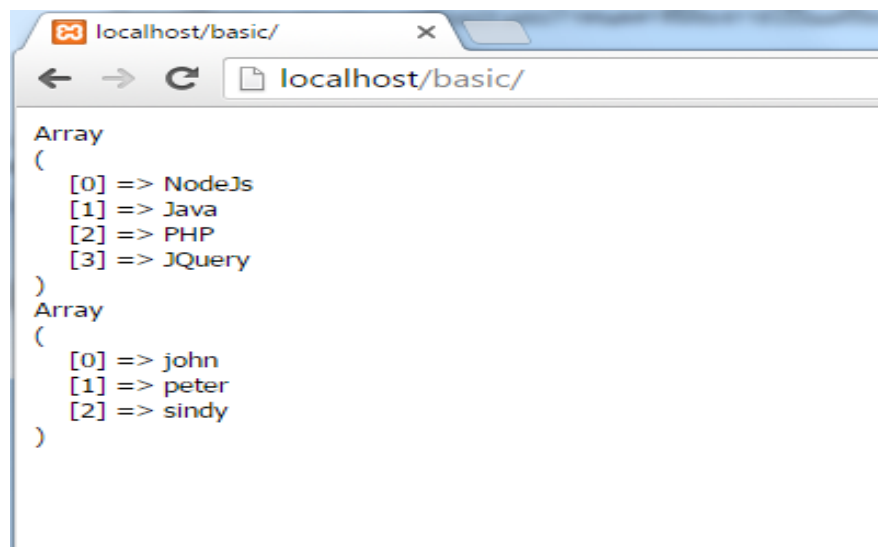
เราสามารถเข้าถึงข้อมูลภายใน Array โดยกำหนด Key ที่ต้องการ ตัวอย่างต้องการแสดงผลค่าของจำนวนหนังสือ ของ Key ที่มีชื่อว่า PHP จะได้ผลลัพธ์เป็น 10

```
echo $book['PHP'];
```

- ในการใช้งานเราจะประกาศตัวแปร และกำหนดค่าที่ต้องการไว้ในเครื่องหมาย [] หรือ array()

```
1 <?php
2
3     $book = ['NodeJs', 'Java', 'PHP', 'jQuery'];
4
5     $name = array('john', 'peter', 'sindy');
6
7     echo"<pre>";
8         print_r($book);
9         print_r($name);
10    echo"</pre>";
11
12 ?>
```

ตัวอย่างการแสดงผล



```
Array
(
    [0] => NodeJs
    [1] => Java
    [2] => PHP
    [3] => jQuery
)
Array
(
    [0] => john
    [1] => peter
    [2] => sindy
)
```

จากตัวอย่างข้างต้น เป็นการใช้งาน Array แบบ 1 มิติ เท่านั้น ยังมี Array แบบ 2 มิติ และ Array แบบ 3 มิติ ที่ยังไม่ได้กล่าวถึงในEBook เล่มนี้

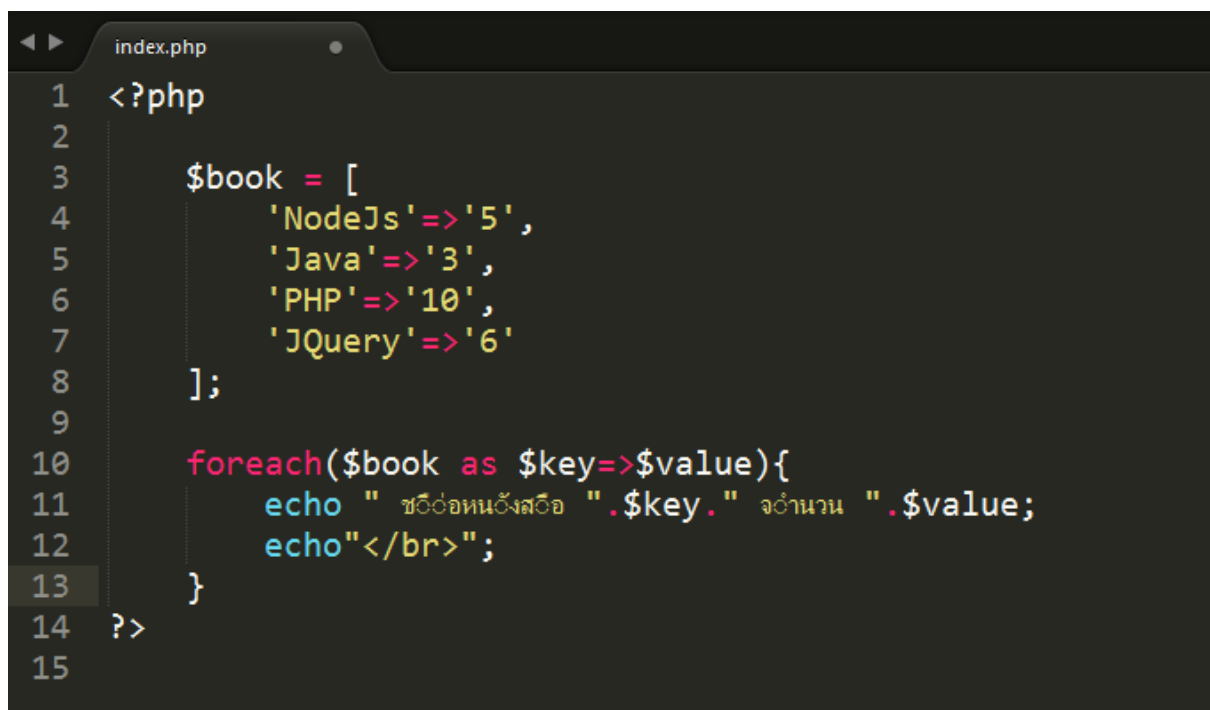
การใช้งานคำสั่ง foreach เพื่อวนลูปแสดงข้อมูลภายใน Array

ตัวอย่างการใช้งาน foreach กรณีต้องการ แสดงค่า Key และ Value

รูปแบบคำสั่ง

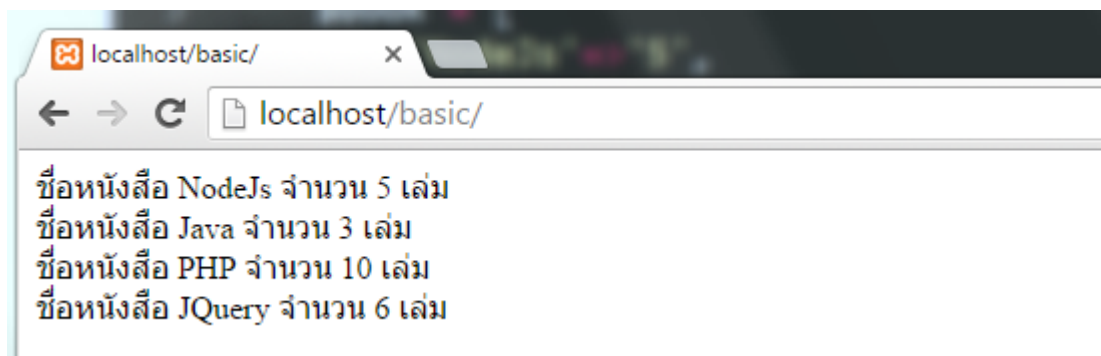
```
foreach ( ตัวแปรที่เป็น Array as ตัวแปรที่รับค่า Key => ตัวแปรที่รับค่า Value){  
    คำสั่งที่ต้องการให้ทำงาน  
}
```

ตัวอย่างคำสั่ง foreach



```
1 <?php  
2  
3 $book = [  
4     'NodeJs'=>'5',  
5     'Java'=>'3',  
6     'PHP'=>'10',  
7     'jQuery'=>'6'  
8 ];  
9  
10 foreach($book as $key=>$value){  
11     echo " ชื่อหนังสือ ".$key." จำนวน ".$value;  
12     echo"<br>";  
13 }  
14 ?>  
15
```

ตัวอย่างการแสดงผล



localhost/basic/

localhost/basic/

ชื่อหนังสือ NodeJs จำนวน 5 เล่ม
ชื่อหนังสือ Java จำนวน 3 เล่ม
ชื่อหนังสือ PHP จำนวน 10 เล่ม
ชื่อหนังสือ JQuery จำนวน 6 เล่ม

กรณีในตัวแปร Array ไม่ได้มีการกำหนด Key Array จะกำหนด Key ให้เป็นตัวเลขโดยอัตโนมัติโดยเริ่มต้นด้วย 0

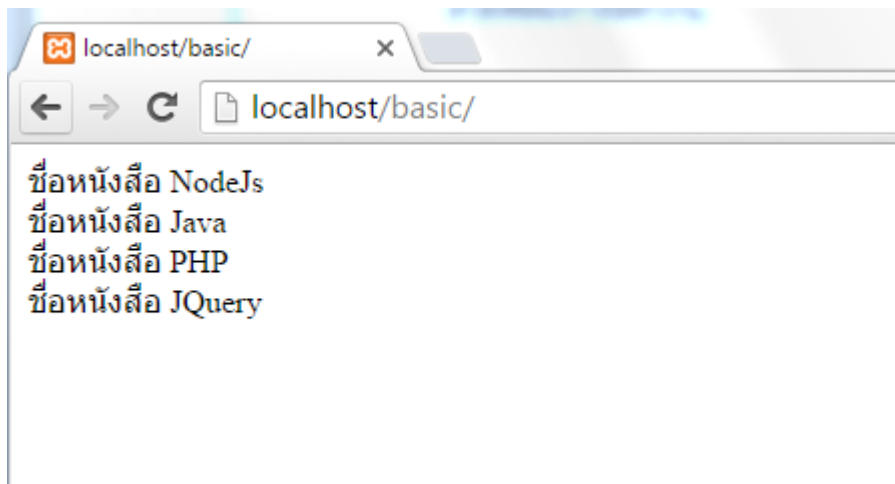
รูปแบบคำสั่ง

```
foreach ( ตัวแปรที่เป็น Array as ตัวแปรที่รับค่า Value){  
    คำสั่งที่ต้องการให้ทำงาน  
}
```

ตัวอย่างคำสั่ง

```
1 <?php  
2  
3     $book = ['NodeJs','Java','PHP','jQuery',];  
4  
5     foreach($book as $value){  
6         echo " ชื่อหนังสือ ".$value;  
7         echo"<br>";  
8     }  
9 ?>
```

ตัวอย่างการแสดงผล



Function

Function คือ การนำคำสั่งที่ใช้บ่อย ๆ มารวมกัน ไว้ สามารถ เรียกใช้งานได้ง่าย และ นำกลับมาใช้ใหม่ โดยไม่ต้องเขียนคำสั่งใหม่ทั้งหมด เรียกแค่ชื่อ function และ อาจมีการส่งค่าไปให้ function ทำงาน และ ส่งผลลัพธ์กลับมา ซึ่ง function มีทั้งแบบที่ PHP มีมาให้เราใช้งานได้อยู่และ กับแบบที่เราต้องการ function ขึ้นมาเองเพื่อให้ตรงกับความต้องการ


ตัวอย่างการเรียกใช้งาน function ที่มีมาอยู่แล้วใน PHP

การเรียกใช้งาน function phpinfo() เพื่อแสดงข้อมูลของ รายละเอียดค่าของ php ที่ใช้งานอยู่



```
1 <?php
2
3 phpinfo();
4
5 ?>
```

ตัวอย่างการทำงาน

PHP Version 5.6.15		
System	Windows NT BANK-PC 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586	
Build Date	Oct 29 2015 12:34:05	
Compiler	MSVC11 (Visual C++ 2012)	
Architecture	x86	
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"	
Server API	Apache 2.0 Handler	

การสร้าง function ขึ้นมาเอง แบบไม่มีการส่งค่าระหว่าง function และการเรียกใช้งาน

```
1 <?php
2
3 function hello(){
4     echo"Hello PHP Web Applicaiton";
5 }
6
7 hello();
8 ?>
9
```

การตั้งชื่อ function นั้น จะต้องไม่ซ้ำกับชื่อ function ที่มีมาอยู่แล้วใน PHP และควรสื่อความหมายการทำงาน

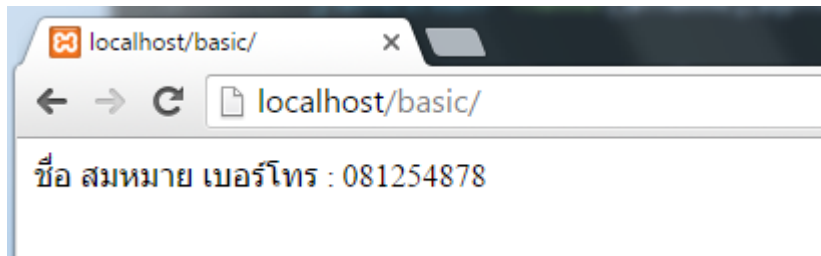
การสร้าง function ที่มีการส่งค่าระหว่าง function

ตัวอย่างเป็นการสร้าง function ชื่อ name และมีการส่งค่าพารามิเตอร์ 2 ตัว คือ ตัวแปร \$name และ ตัวแปร \$phone

```
1 <?php
2
3 function name($name,$phone){
4     echo "ชื่อ ".$name." เบอร์โทร : ".$phone;
5 }
6
7
8 name("สมหมาย","081254878");
9 ?>
10
```

การเรียกใช้งาน เราต้องระบุค่าของ พารามิเตอร์ที่ต้องการนำไปใช้งานใน function

ตัวอย่างการแสดงผล

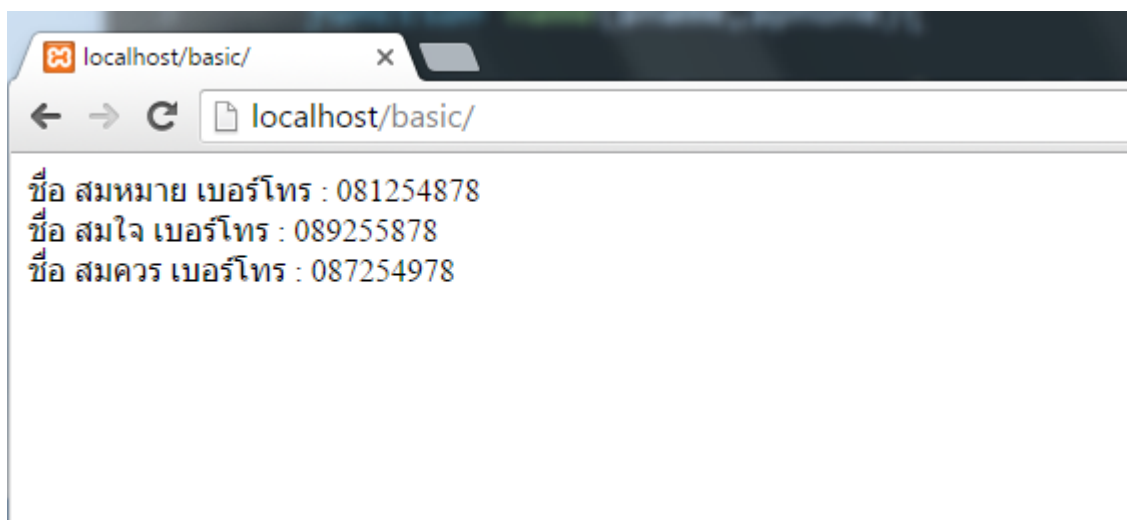


ข้อดีของ function คือเราสามารถเรียกใช้งาน function ซ้ำหลาย ๆ ครั้งได้โดยไม่ต้องเขียน Code เดิมอีก เพียงแค่เปลี่ยนข้อมูลใน พารามิเตอร์ตอนที่เรียกใช้งาน

ตัวอย่างการเรียกใช้งาน Function ซ้ำกันหลาย ๆ ครั้ง



ตัวอย่างการแสดงผล



การใช้งาน PHP MySQLi ในการจัดการฐานข้อมูล MySQL

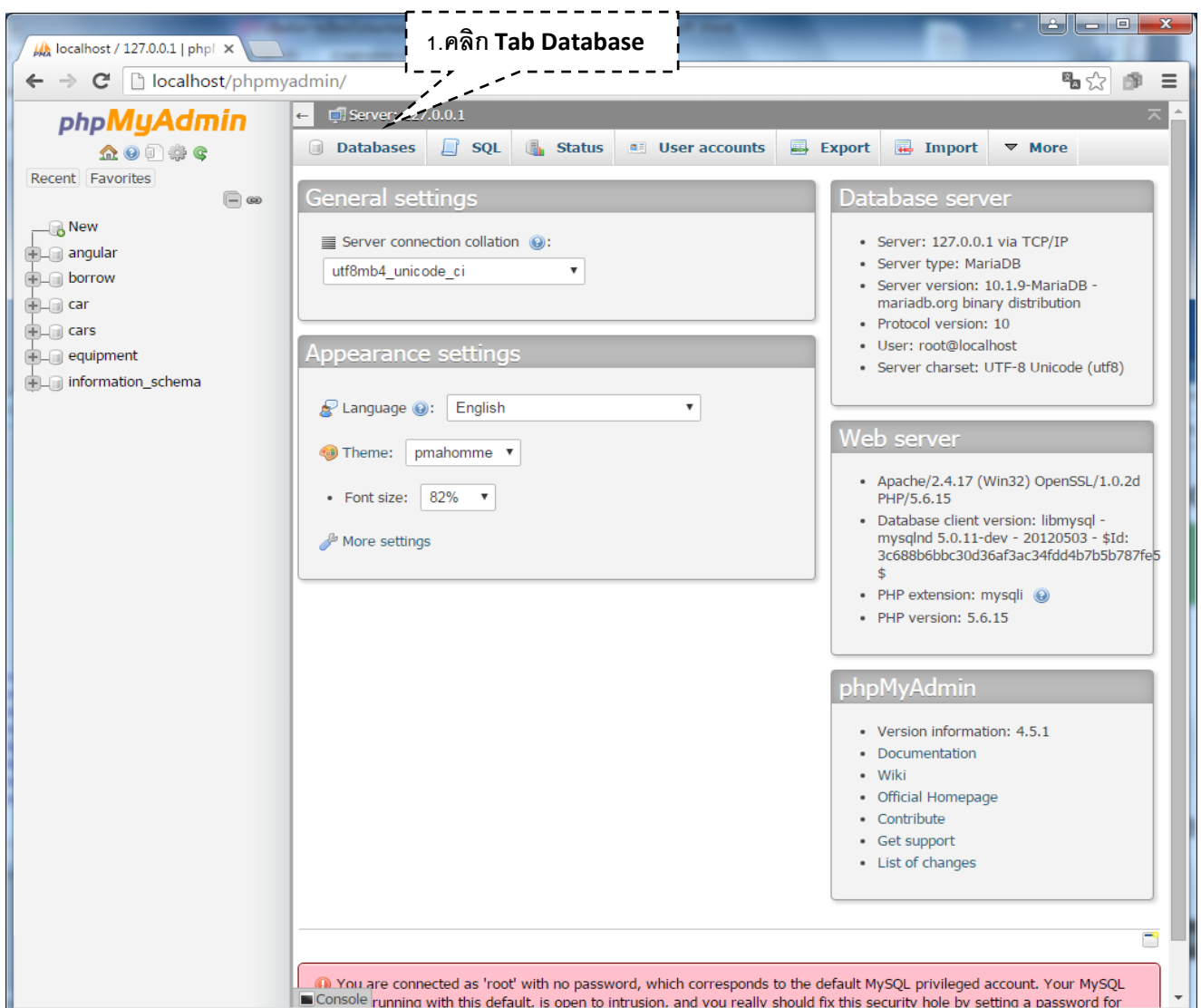
ในการเขียนโปรแกรม PHP เพื่อจัดการฐานข้อมูล ใน PHP นั้น สามารถเขียนได้ 2 แบบ คือ แบบ mysqli กับ แบบ PHP PDO ซึ่งในการเขียนโปรแกรมเชื่อมต่อกับฐานข้อมูลนั้น ผู้อ่านจำเป็นต้องมีความรู้ใน ภาษา SQL (Structured Query Language) ซึ่งเป็นภาษามาตรฐานสำหรับจัดการข้อมูลในฐานข้อมูล เช่น การ ค้นหา เพิ่ม แก้ไข และ ลบข้อมูล ในฐานข้อมูล

ความแตกต่างระหว่างการเขียน mysqli กับ PDO คือ

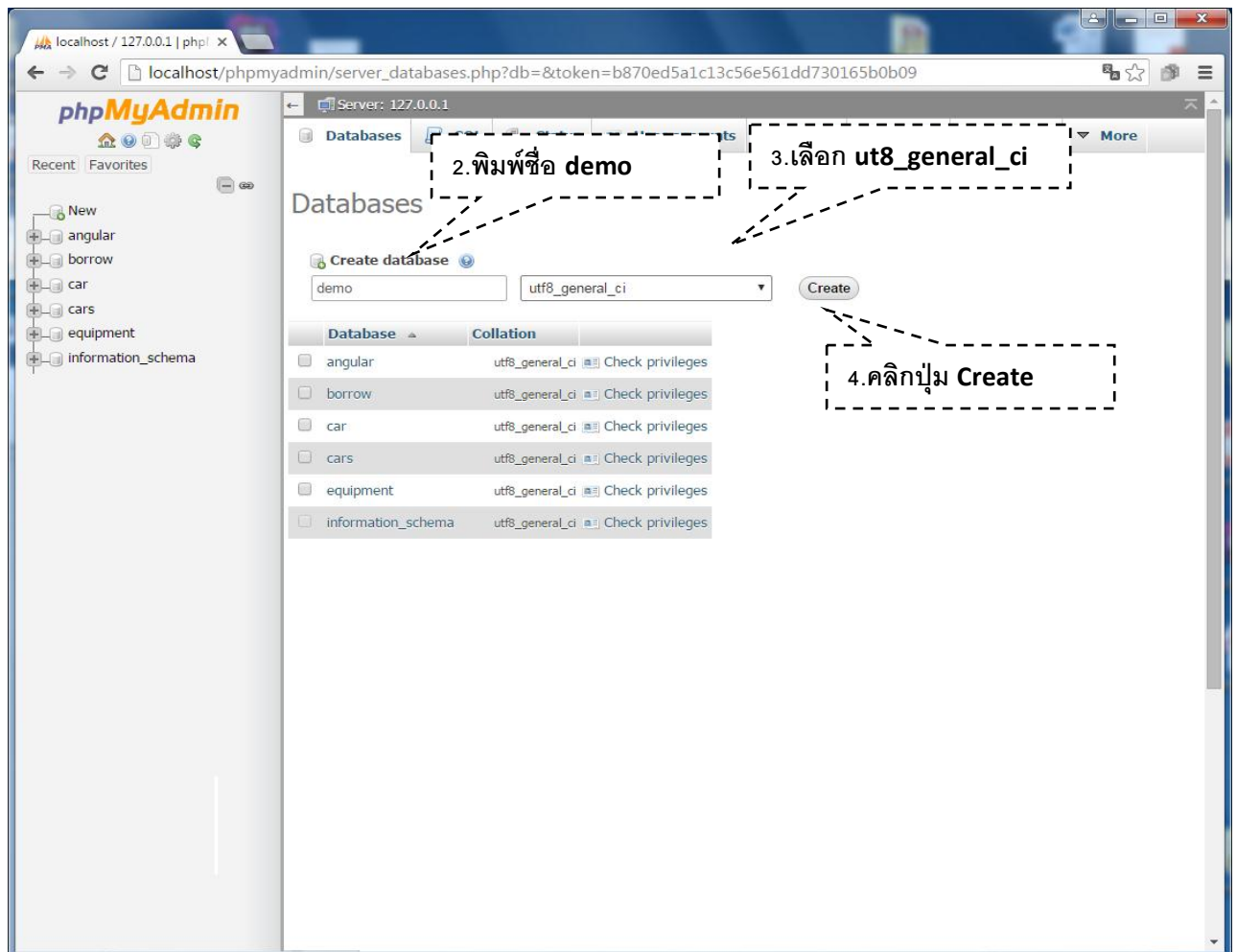
การเขียนแบบ php mysqli จะรองรับเฉพาะฐานข้อมูล MySQL เท่านั้น ส่วนการเขียน PHP แบบ PDO จะรองรับการเชื่อมต่อฐานข้อมูลที่หลากหลาย
ซึ่งใน Ebook เล่มนี้จะแนะนำการใช้งาน PHP แบบ mysqli ในการจัดการฐานข้อมูล โดยใช้ร่วมกับ PhpMyAdmin

การสร้างฐานข้อมูลใน PhpMyAdmin

เมื่อติดตั้งโปรแกรม XAMPP สำเร็จแล้ว สามารถเข้าสู่ระบบจัดการฐานข้อมูลได้โดยพิมพ์ localhost /phpmyadmin เมื่อเข้าสู่หน้าจอ phpmyadmin แล้วคลิกที่ เมนู Database



ในช่อง Create database พิมพ์ชื่อฐานข้อมูลที่ต้องการสร้าง และในช่อง Collation เลือกเป็น utf8_general_ci เพื่อรองรับกับภาษาไทย



เมื่อสร้างฐานข้อมูลเรียบร้อยแล้ว เราจะได้ฐานข้อมูลชื่อว่า demo ซึ่งจะยังไม่มีข้อมูลอะไร เราต้องมาทำการสร้างตารางเพื่อเก็บข้อมูล ต่าง ๆ ที่ต้องใช้ในระบบของเรา ซึ่งในการสร้างตารางนั้นเราจำเป็นต้องออกแบบโครงสร้างของตารางเสียก่อนที่จะเริ่มลงมือเขียนโปรแกรม แต่ในตัวอย่างนี้จะเป็นการสร้างตาราง employees แบบง่าย ๆ เก็บเพียง ชื่อ ที่อยู่ เงินเดือน วันเกิด เพื่อแสดงให้เห็นถึงการเขียนโปรแกรม เพื่อค้นหา เพิ่ม แก้ไข ลบ ข้อมูล ในฐานข้อมูล

การสร้างตาราง employees ด้วยคำสั่ง CREATE TABLE

ตัวอย่างคำสั่งสร้าง ตาราง employees

```
1  
2 CREATE TABLE employees (  
3     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
4     fullname VARCHAR(100) NOT NULL,  
5     address VARCHAR(250) NOT NULL,  
6     salary INT(11) NOT NULL,  
7     birthday DATE NOT NULL  
8 );
```

อธิบาย คำสั่ง SQL เพิ่มเติม

บรรทัดที่ 2 สร้างตารางชื่อว่า employees

บรรทัดที่ 3 สร้าง คอลัมน์ ชื่อว่า id เก็บข้อมูลเป็น INT (ตัวเลข) ไม่เป็นค่าว่าง NOT NULL เพิ่มลำดับเลข คอลัมน์อัตโนมัติ AUTO_INCREMENT และ กำหนดให้เป็น PRIMARY KEY คือ เป็น คีย์หลัก สามารถ อ้างอิงถึงข้อมูลอื่น ๆ ในตารางได้

บรรทัดที่ 4 สร้าง คอลัมน์ชื่อว่า fullname เก็บค่าเป็น VARCHAR (100) คือ ค่าสตริงความยาวไม่เกิน 100 ตัวอักษร

บรรทัดที่ 5 สร้าง คอลัมน์ชื่อว่า address เก็บค่าเป็น VARCHAR (250) คือ ค่าสตริงความยาวไม่เกิน 250 ตัวอักษร

บรรทัดที่ 6 สร้าง คอลัมน์ชื่อว่า salary เก็บค่าเป็น INT(11) คือ ค่าตัวเลขความยาวไม่เกิน 11 หลัก

บรรทัดที่ 7 สร้าง คอลัมน์ชื่อว่า birthday เก็บค่าเป็น DATE คือ เก็บค่าวันที่ในรูปแบบ ปี(ค.ศ.) – เดือน – วัน

คลิกที่เมนู SQL เพิ่มคำสั่งสำหรับการสร้างตารางดังนี้

2.คลิก Tab SQL

1.คลิก Database Demo

4.คลิก ปุ่ม Go

3. เพิ่ม คำสั่ง สร้างตาราง employees

```
1  
2 CREATE TABLE employees (  
3   id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
4   fullname VARCHAR(100) NOT NULL,  
5   address VARCHAR(250) NOT NULL,  
6   salary INT(11) NOT NULL,  
7   birthday DATE NOT NULL  
8 );
```

เมื่อสร้างตารางเรียบร้อยแล้ว

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.2296 seconds.)

```
CREATE TABLE employees ( id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, fullname VARCHAR(100) NOT NULL, address VARCHAR(250) NOT NULL, salary INT(11) NOT NULL, birthday DATE NOT NULL )
```

[Edit inline] [Edit] [Create PHP code]

การเพิ่มข้อมูลเข้าสู่ตาราง ด้วยคำสั่ง INSERT

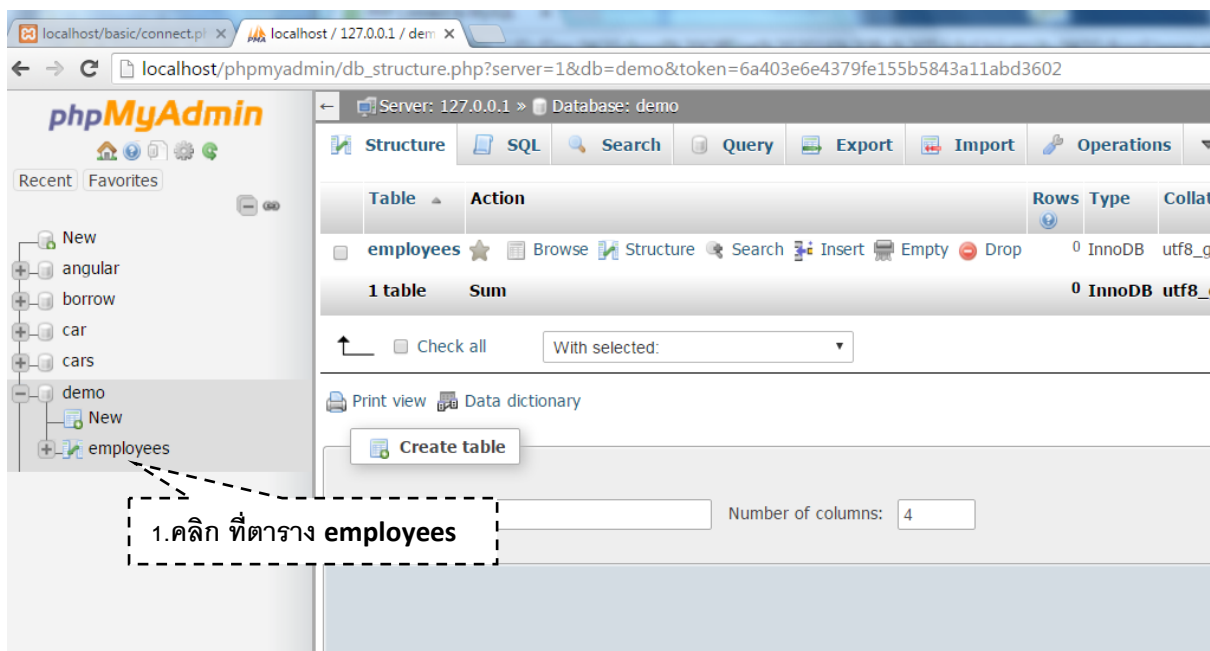
ในตัวอย่างเราจะทำการเพิ่มข้อมูลเข้าไปในตาราง employee ก่อน ด้วยคำสั่ง INSERT โดยคำสั่งมีรูปแบบการเขียนดังนี้

รูปแบบการเขียนคำสั่ง INSERT

```
INSERT INTO ชื่อตาราง (คอลัมน์1,คอลัมน์2,คอลัมน์3,คอลัมน์4,คอลัมน์5)  
VALUES (ข้อมูล1,ข้อมูล2,ข้อมูล3,ข้อมูล4,ข้อมูล5);
```

ตัวอย่างคำสั่งที่ใช้เพิ่มข้อมูลลงในตาราง employees

```
INSERT INTO employees (id , fullname , address , salary , birthday ) VALUES  
(NULL, 'สมชาย ใจดี', '105 แขวง บางมด เขต พุ่งครุ กรุงเทพฯ 10140', '2500', '1987-06-28'),  
(NULL, 'สมหญิง สายเสมอ', '269/10 ซ.ลาดพร้าว ถนน ลาดพร้าว แขวงวังทองหลาง เขต วังทองหลาง  
กรุงเทพ 10310', '12000', '1980-04-26');
```



localhost/basic/connect.pl X localhost / 127.0.0.1 / dem X

localhost/phpmyadmin/tbl_sql.php?db=demo&table=employees&token=6a403e6e4379fe155b5843a11abd3602

Server: 127.0.0.1 » Database: demo » Table: employees

Browse Structure SQL Search Insert Export Import More

Run SQL query/queries on table

1.คลิก ที่ Tab SQL

Columns

id
fullname
address
salary
birthday

1 INSERT INTO `employees` (`id`, `fullname`, `address`, `salary`, `birthday`) VALUES (NULL, 'สมชาย ใจดี', '105 แขวง บางมด เขต ท่งครุ กรุงเทพฯ 10140', '2500', '1987-06-28'), (NULL, 'สมหญิง สายเสมอ', '269/10 ซ.ลาดพร้าว ถนน ลาดพร้าว แขวง วังทองหลาง เขต วังทองหลาง กรุงเทพฯ 10310', '12000', '1980-04-26');

2.เพิ่มคำสั่ง

INSERT INTO employees (id , fullname , address , salary , birthday) VALUES (NULL, 'สมชาย ใจดี', '105 แขวง บางมด เขต ท่งครุ กรุงเทพฯ 10140', '2500', '1987-06-28'), (NULL, 'สมหญิง สายเสมอ', '269/10 ซ.ลาดพร้าว ถนน ลาดพร้าว แขวงวังทองหลาง เขต วังทองหลาง กรุงเทพฯ 10310', '12000', '1980-04-26');

SELECT*
DELETE
☐ Bind parameter

Bookmark this SQL query:

[Delimiter ;] ☒ Show this query here again ☐ Retain query box ☐ Rollback when finished
☒ Enable foreign key checks

Go

3.คลิกปุ่ม Go

localhost/basic/connect.pl X localhost / 127.0.0.1 / dem X

localhost/phpmyadmin/tbl_sql.php?db=demo&table=employees&token=6a403e6e4379fe155b5843a11abd3602

Server: 127.0.0.1 » Database: demo » Table: employees

Browse Structure SQL Search Insert Export Import More

Privileges More

Show query box

2 rows inserted.
Inserted row id: 2 (Query took 0.0400 seconds.)

INSERT INTO `employees` (`id`, `fullname`, `address`, `salary`, `birthday`) VALUES (NULL, 'สมชาย ใจดี', '105 แขวง บางมด เขต ท่งครุ กรุงเทพฯ 10140', '2500', '1987-06-28'), (NULL, 'สมหญิง สายเสมอ', '269/10 ซ.ลาดพร้าว ถนน ลาดพร้าว แขวงวังทองหลาง เขต วังทองหลาง กรุงเทพฯ 10310', '12000', '1980-04-26')

[Edit inline] [Edit] [Create PHP code]

4.เมื่อเพิ่มข้อมูลเรียบร้อย

localhost/basic/connect.php x localhost / 127.0.0.1 / dem x

localhost/phpmyadmin/sql.php?db=demo&table=employees&token=6a403e6e4379fe155b5843a11abd3602&pos=0

Server: 127.0.0.1 » Database: demo » Table: employees

Browse Structure SQL Search Insert Export Import Privileges More

Showing rows (total, Query took 0.0010 seconds.)

1.คลิก Tab Browse

SELECT * FROM

[Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Sort by key: None

2.จะแสดงข้อมูลที่เพิ่มเข้ามา

			id	fullname	address	salary	birthday	
<input type="checkbox"/>	Edit	Copy	Delete	1	สมชาย ใจดี	105 แขวง บางมด เขต ทุ่งครุ กรุงเทพฯ 10140	2500	1987-06-28
<input type="checkbox"/>	Edit	Copy	Delete	2	สมหญิง สายเสมอ	269/10 ซ.ลาดพร้าว ถนน ลาดพร้าว แขวงวังทองหลาง เขต ...	12000	1980-04-26

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print view Export Display chart Create view

Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Console

การเชื่อมต่อฐานข้อมูล PHP MySQLi

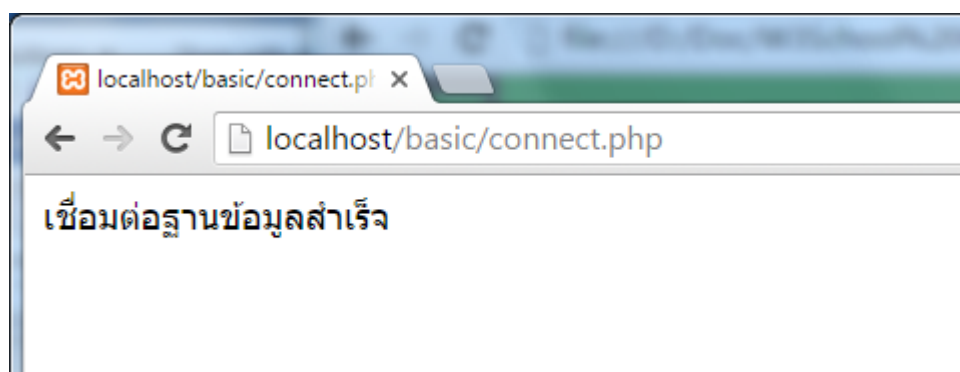
การเชื่อมต่อฐานข้อมูล MySQL เราจะใช้ Function `mysqli_connect` และ กำหนดค่าการเชื่อมต่อฐานข้อมูลเช่น ชื่อ Server Name , Username , Password และชื่อฐานข้อมูล ลงไปใน Function สร้างไฟล์ `connect.php` ลงใน Folder Project ของเรา

- 1.สร้าง Folder ชื่อ basic เก็บไว้ที่ Drive (ที่ติดตั้งโปรแกรม XAMPP เช่น C://xampp/htdocs/basic)
- 2.สร้างไฟล์ `connect.php` เก็บไว้ที่ C://xampp/htdocs/basic/connect.php

ตัวอย่าง C://xampp/htdocs/basic/connect.php กำหนดค่าการเชื่อมต่อฐานข้อมูลดังภาพ

```
connect.php
1 <?php
2
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $db = "demo";
7
8 $conn = mysqli_connect($servername,$username,$password,$db);
9
10 mysqli_set_charset($conn,"utf8");
11
12 if(!$conn){
13     die(" ไม่สามารถเชื่อมต่อฐานข้อมูลได้ " .mysqli_connect_error());
14 }else{
15
16     echo "เชื่อมต่อฐานข้อมูลสำเร็จ";
17
18 }
19
20
21
22 ?>
```

ทดสอบการทำงาน เปิด Web Browser พิมพ์ localhost/basic/connect.php หากเชื่อมต่อฐานข้อมูลสำเร็จจะแสดงหน้าจอ ดังภาพ



การเรียกดูข้อมูล ในฐานข้อมูล

1. ก่อนอื่นจะต้องสร้างไฟล์ index.php ขึ้นมา และ นำเข้าไฟล์ connect.php ด้วยคำสั่ง require() เพื่อเชื่อมต่อกับฐานข้อมูล (ในตัวอย่างจะแสดงการเขียน Source Code ในไฟล์ index.php ที่ละขั้นตอน)

```
index.php
1  <?php
2      require("connect.php");
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <title>PHP Basic</title>
8  </head>
9      <body>
10
11
12      </body>
13  </html>
```

2. สร้างตารางด้วย แท็ก table สำหรับเตรียมนำข้อมูลมาแสดง

```
index.php
1  <?php
2      require("connect.php");
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <title>PHP Basic</title>
8  </head>
9  <body>
10      <table border="1">
11          <tr>
12              <th width="5%">ลำดับ</th>
13              <th width="15%">ชื่อ - นามสกุล</th>
14              <th width="60%">ที่อยู่</th>
15              <th width="10%">เงินเดือน</th>
16              <th width="10%">วันเกิด</th>
17          </tr>
18      </table>
19
20      </body>
21  </html>
22
23
24
```

3. สร้างตัวแปร \$sql เก็บคำสั่ง SQL เรียกดูข้อมูลจากตาราง employees เรียกใช้ ฟังก์ชัน mysqli_query โดยรับค่าตัวแปร \$conn (มาจากไฟล์ connect.php) และ ตัวแปร \$sql และ นำผลที่ได้ มาเก็บไว้ในตัวแปร \$result

```
index.php
1 <?php
2     require("connect.php");
3 >
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <title>PHP Basic</title>
8 </head>
9 <body>
10
11     <table border="1">
12         <tr>
13             <th width="5%">ลำดับ</th>
14             <th width="15%">ชื่อ - นามสกุล</th>
15             <th width="60%">ที่อยู่</th>
16             <th width="10%">เงินเดือน</th>
17             <th width="10%">วันเกิด</th>
18         </tr>
19         <?php
20
21             $sql = "SELECT * FROM employees";
22
23             $result = mysqli_query($conn,$sql);
24
25         ?>
26     </table>
27
28 </body>
29 </html>
```

4. สร้างตัวแปร \$i เพื่อใช้แสดงลำดับแสดงข้อมูล และ ใช้คำสั่ง if ตรวจสอบการทำงานของ ฟังก์ชัน mysqli_num_rows (\$result) ว่ามีค่ามากกว่า 0 หรือไม่

```
15         <th width="60%">ที่อยู่</th>
16         <th width="10%">เงินเดือน</th>
17         <th width="10%">วันเกิด</th>
18     </tr>
19     <?php
20         $sql = "SELECT * FROM employees";
21
22         $result = mysqli_query($conn,$sql);
23         $i = 1;
24
25         if(mysqli_num_rows($result) > 0){
26
27
28
29         }else{
30
31             echo "EMPTY DATA";
32         }
33     ?>
34 </table>
35
36 </body>
37 </html>
```

5. ใช้คำสั่ง while เพื่อนำข้อมูล ที่ได้จากฟังก์ชัน mysqli_fetch_assoc มาเก็บไว้ที่ตัวแปร \$row แสดงผลข้อมูลจากฐานข้อมูลที่อยู่ตัวแปร \$row ตามชื่อ คอลัมน์ในตาราง employees และเพิ่มค่า \$i ที่ละ 1

```
<?php
$sql = "SELECT * FROM employees";

$result = mysqli_query($conn,$sql);
$i = 1;

if(mysqli_num_rows($result) > 0){

    while($row = mysqli_fetch_assoc($result)){
        echo"<tr>";
        echo"<td>".$i."</td>";
        echo"<td>".$row['fullname']."</td>";
        echo"<td>".$row['address']."</td>";
        echo"<td style='text-align:right;'>".number_format($row['salary'])."</td>";
        echo"<td>".$row['birthday']."</td>";
        echo"<td><a href='edit_form.php?id=".$row['id']."'>แก้ไข</a></td>";

        echo"<td><a href='data.php?delete_id=".$row['id']."'
        onclick='return confirm(\"คุณต้องการลบข้อมูลหรือไม่\")'>ลบ</a>";

        echo "</td>";

        echo "</tr>";

        $i++;
    }

}else{

    echo "EMPTY DATA";
}
?>
```

6. ทดสอบการทำงานของ ฟังก์ชัน localhost/basic/index.php ใน Web Browser

PHP Basic

localhost/basic/

ข้อมูลพนักงาน

เพิ่มข้อมูล

ลำดับ	ชื่อ - นามสกุล	ที่อยู่	เงินเดือน	วันเกิด	แก้ไข	ลบ
1	สมชาย ใจดี	105 แขวง บางมด เขต ทungkri กรุงเทพฯ 10140	9,999	1987-06-28	แก้ไข	ลบ
2	สมหญิง สายเสมอ	269/10 ซ.ลาดพร้าว ถนน ลาดพร้าว แขวงวังทองหลาง เขต วังทองหลาง กรุงเทพฯ 10310	12,000	1980-04-26	แก้ไข	ลบ

ตัวอย่าง SourceCode ไฟล์ index.php ทั้งหมด

```
connect.php x edit_form.php data.php x index.php add_form.php x
1 <?php
2     require("connect.php");
3 ?>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <title>PHP Basic</title>
8 </head>
9 <body>
10    <h1>ข้อมูลพนักงาน</h1>
11    <a href="add_form.php">เพิ่มข้อมูล</a>
12    <table border="1">
13        <tr>
14            <th width="5%">ลำดับ</th>
15            <th width="15%">ชื่อ - นามสกุล</th>
16            <th width="50%">ที่อยู่</th>
17            <th width="10%">เงินเดือน</th>
18            <th width="10%">วันเกิด</th>
19            <th width="5%">แก้ไข</th>
20            <th width="5%">ลบ</th>
21        </tr>
22    <?php
23        $sql = "SELECT * FROM employees";
24        $result = mysqli_query($conn,$sql);
25        $i = 1;
26        if(mysqli_num_rows($result) > 0){
27
28            while($row = mysqli_fetch_assoc($result)){
29                echo"<tr>";
30                echo"<td>".$i."</td>";
31                echo"<td>".$row['fullname']."</td>";
32                echo"<td>".$row['address']."</td>";
33                echo"<td style='text-align:right;'>".number_format($row['salary'])."</td>";
34                echo"<td>".$row['birthday']."</td>";
35                echo"<td><a href='edit_form.php?id=".$row['id']."'>แก้ไข</a></td>";
36                echo"<td><a href='data.php?delete_id=".$row['id']."'";
37                echo"onclick='return confirm(\"คุณต้องการลบข้อมูลหรือไม่\")'>ลบ</a></td>";
38                echo "</tr>";
39                $i++;
40            }
41        }else{
42            echo "EMPTY DATA";
43        }
44    ?>
45    </table>
46 </body>
47 </html>
```


การสร้างฟอร์มบันทึกข้อมูล

การสร้างฟอร์มบันทึกข้อมูล ในตัวอย่าง จะทำการสร้างไฟล์ add_form.php สำหรับ สร้างฟอร์ม และ สร้างไฟล์ data.php สำหรับรับข้อมูลจากฟอร์ม บันทึกเข้าสู่ฐานข้อมูล

ตัวอย่างไฟล์ add_form.php

```
1
2 <!DOCTYPE html>
3 <html lang="th">
4 <head>
5   <title>Bootstrap Example</title>
6
7 </head>
8
9 <body>
10
11 <form action="data.php" method="post">
12   <fieldset>
13
14     <legend>Personal information:</legend>
15
16     <label>ชื่อย่อ - นามสกุล:</label><br>
17
18     <input type="text" name="fullname" ><br>
19
20     <label> ที่อยู่:</label><br>
21
22     <textarea name="address" rows="5" cols="30"></textarea><br>
23
24     <label> เงินเดือน:</label><br>
25
26     <input type="number" name="salary" ><br>
27
28     <label> วันเกิด :</label><br>
29
30     <input type="date" name="birthday" ><br><br>
31
32     <input type="submit" value="บันทึกข้อมูล">
33
34   </fieldset>
35
36 </form>
37
38 </body>
39
40 </html>
```

ตัวอย่างไฟล์ data.php

ในตัวอย่างจะ ใช้เป็นการรับค่าข้อมูลมาจาก ไฟล์ add_form.php ในรูปแบบ POST

```
connect.php x index.php x add_form.php x data.php
1 <?php
2
3     require("connect.php");
4
5     //add new data form to mysql
6     if($_POST['fullname']){
7
8         $fullname = $_POST['fullname'];
9
10        $address = $_POST['address'];
11
12        $salary = $_POST['salary'];
13
14        $birthday = $_POST['birthday'];
15
16        $sql = "INSERT INTO employees ( id , fullname, address, salary, birthday)
17
18            VALUES (null , '$fullname', '$address' , '$salary' , '$birthday' )";
19
20        if (mysqli_query($conn, $sql)) {
21
22            echo "New record created successfully";
23
24            echo "<br><a href='index.php'>Bact To All Employees Page</a>";
25        } else {
26
27            echo "Error: " . $sql . "<br>" . mysqli_error($conn);
28        }
29    }
30
31
32
33 ?>
```

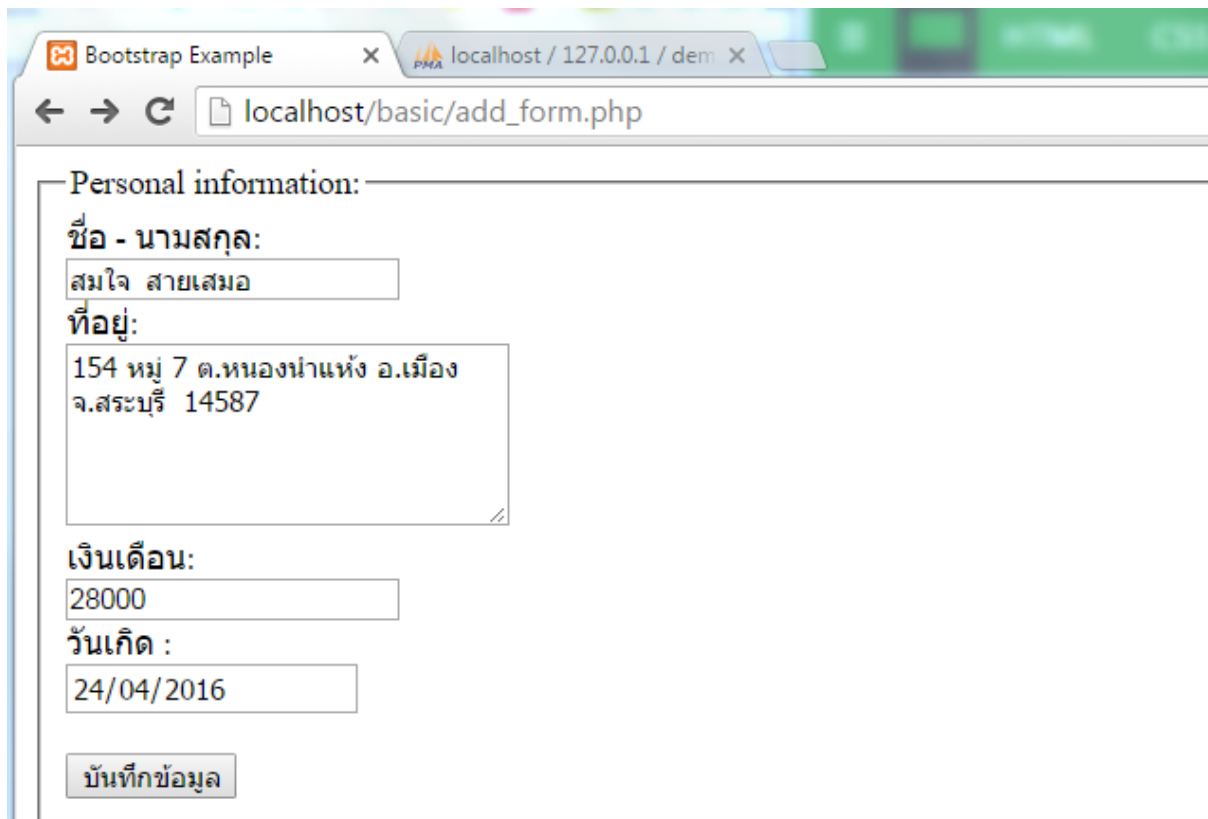
บรรทัดที่ 3 ใช้คำสั่ง require() นำเข้าไฟล์ connect.php เพื่อเชื่อมต่อฐานข้อมูล

บรรทัดที่ 6 ใช้คำสั่ง if ตรวจสอบ ค่าในตัวแปรรูปแบบ POST ว่ามี ค่า Key เท่ากับ fullname ส่งมาหรือไม่

บรรทัดที่ 8 – 14 รับค่าจากตัวแปร POST และ เก็บข้อมูลลงในตัวแปรของแต่ละ คอลัมน์

บรรทัดที่ 16 ตัวแปร \$sql เก็บค่าคำสั่งสำหรับ เพิ่มข้อมูลลงฐานข้อมูล

ทดสอบการเพิ่มข้อมูล ไปที่ URL : localhost/basic/add_form.php กรอกข้อมูลตัวอย่างเพิ่มทดสอบการส่งข้อมูล



Bootstrap Example x localhost / 127.0.0.1 / dem x

localhost/basic/add_form.php

Personal information:

ชื่อ - นามสกุล:
สมใจ สายเสมอ

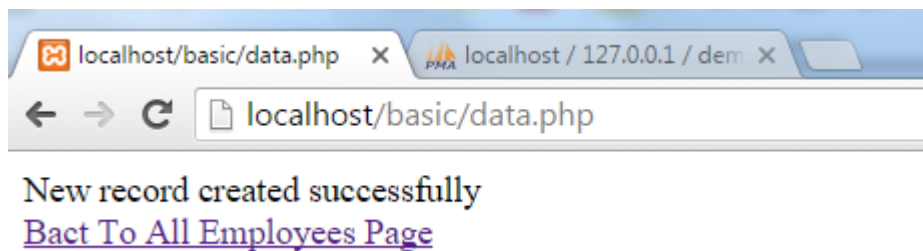
ที่อยู่:
154 หมู่ 7 ต.หนองน้ำแห้ง อ.เมือง
จ.สระบุรี 14587

เงินเดือน:
28000

วันเกิด :
24/04/2016

บันทึกข้อมูล

หากบันทึกข้อมูลสำเร็จจะแสดงผลดังภาพ



ตรวจสอบใน localhost/phpmyadmin จะพบข้อมูลที่เรบันทึกเข้าไปใหม่

+ Options

	id	fullname	address	salary	birthday
<input type="checkbox"/> Edit Copy Delete	1	สมชาย ใจดี	105 แขวง บางมด เขต ท่งครุ กรุงเทพฯ 10140	2500	1987-06-28
<input type="checkbox"/> Edit Copy Delete	2	สมหญิง สายเสมอ	269/10 ซ.ลาดพร้าว ถนน ลาดพร้าว แขวงวังทองหลาง เขต ...	12000	1980-04-26
<input type="checkbox"/> Edit Copy Delete	3	สมใจ สายเสมอ	154 หมู่ 7 ต.หนองน้ำแห้ง อ.เมือง จ.สระบุรี 14587	28000	2016-04-24

Check all With selected: Edit Copy Delete Export

การสร้างฟอร์มแก้ไขข้อมูล

สร้างไฟล์ edit_form.php สำหรับแก้ไขข้อมูล โดยจะรับค่า \$_GET['id'] เพื่อนำไปค้นหาข้อมูลที่อยู่ในฐานข้อมูล และนำมา แสดงใน ฟอร์ม เพื่อทำการแก้ไข

ตัวอย่างปุ่ม แก้ไข ในไฟล์ index.php ส่งค่า id รูปแบบ GET ไปที่ไฟล์ edit_form.php

```
echo"<td><a href='edit_form.php?id=".$row['id']."'>แก้ไข</a></td>";
```

ตัวอย่างไฟล์ edit_form.php



```
1 <?php
2 require("connect.php");
3
4 if(isset($_GET['id'])){
5
6     $id = $_GET['id'];
7
8     $sql = "SELECT * FROM employees WHERE id = $id";
9
10    $row = mysqli_query($conn, $sql);
11
12    $result = mysqli_fetch_assoc($row);
13
14    if(!$result){
15
16        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
17    }
18 }
19
20 ?>
```

บรรทัดที่ 2 คำสั่ง require () นำเข้าไฟล์ connect.php สำหรับการเชื่อมต่อฐานข้อมูล

บรรทัดที่ 4 ในคำสั่ง if ใช้ ฟังก์ชัน isset ตรวจสอบ ว่า มีการส่งค่า ตัวแปร \$_GET['id'] มาหรือไม่

บรรทัดที่ 5 สร้างตัวแปร \$id รับค่ามาจาก ตัวแปร \$_GET['id']

บรรทัดที่ 8 สร้างตัวแปร \$sql เก็บค่าคำสั่ง SQL ค้นหา ข้อมูล ที่มีค่า id เท่ากับ ตัวแปร \$id

บรรทัดที่ 10 สร้าง ตัวแปร \$row เก็บค่าที่ได้ จาก ฟังก์ชัน mysqli_query()

บรรทัดที่ 12 สร้างตัวแปร \$result เก็บค่าที่ได้จาก ฟังก์ชัน mysqli_fetch_assoc()

บรรทัดที่ 14 คำสั่ง if ตรวจสอบ ว่า ตัวแปร ! \$result มีไม่ค่าเป็นจริงหรือไม่ หากไม่เป็นจริง ให้แสดงผล Error

ตัวอย่างฟอร์มแก้ไขข้อมูล edit_form.php (ต่อ)

```
22 <!DOCTYPE html>
23 <html lang="th">
24 <head>
25   <title>Edit Personal information</title>
26
27 </head>
28 <body>
29
30   <form action="data.php" method="post">
31
32     <fieldset>
33       <legend>Personal information:</legend>
34
35       <label>ชื่อ - นามสกุล:</label><br>
36       <input type="text" name="edit_fullname" required value="<?php echo $result['fullname']?>"><br>
37
38       <label> ที่อยู่:</label><br>
39       <textarea name="edit_address" rows="5" cols="30" required><?php echo $result['address']?></textarea><br>
40
41       <label> เงินเดือน:</label><br>
42       <input type="number" name="edit_salary" value="<?php echo $result['salary']?>"><br>
43
44       <label> วันเกิด:</label><br>
45       <input type="date" name="edit_birthday" value="<?php echo $result['birthday']?>"><br><br>
46
47       <input type="hidden" name="edit_form_id" value="<?php echo $result['id']?>">
48
49       <input type="submit" value="Save">
50
51     </fieldset>
52
53   </form>
54
55 </body>
56 </html>
```

บรรทัดที่ 36 แท็ก input กำหนด name ชื่อ edit_fullname และ value เท่ากับ ตัวแปร \$result['fullname']

บรรทัดที่ 39 แท็ก textarea กำหนด name ชื่อ edit_address และ กำหนดค่า ภายใน เท่ากับ

ตัวแปร \$result['address']

บรรทัดที่ 42 แท็ก input กำหนด name ชื่อ edit_salary และ value มีค่าเท่ากับ \$result['salary']

บรรทัดที่ 45 แท็ก input กำหนด name ชื่อ edit_birthday และ value เท่ากับตัวแปร \$result['birthday']

บรรทัดที่ 47 แท็ก input กำหนดให้มี attribute type มีค่า เท่ากับ hidden และ attribute name ชื่อ edit_form_id และ กำหนดค่า value มีค่าเท่ากับ ค่าของข้อมูล ในตัวแปร \$result['id']

เพิ่มเติม SourceCode ในไฟล์ data.php สำหรับ การแก้ไขข้อมูล

```
33 //save edit data
34
35 if(isset($_POST['edit_form_id'])){
36     $fullname = $_POST['edit_fullname'];
37     $address = $_POST['edit_address'];
38     $salary = $_POST['edit_salary'];
39     $birthday = $_POST['edit_birthday'];
40     $id = $_POST['edit_form_id'];
41
42     $sql = "UPDATE employees SET fullname='$fullname' , address='$address' ,
43         salary = '$salary' ,birthday='$birthday' WHERE id=$id";
44
45     if (mysqli_query($conn, $sql)) {
46         echo "Record updated successfully";
47         echo "<br><a href='index.php'>Bact To All Employees Page</a>";
48     } else {
49         echo "Error updating record: " . mysqli_error($conn);
50     }
51
52 }
53
54
55
56
57
58
59
60
61
62
```

บรรทัดที่ 35 ใช้คำสั่ง if และ คำสั่ง isset ตรวจสอบว่า มีค่าตัวแปร \$_POST['edit_form_id'] ส่งมาหรือไม่

บรรทัดที่ 37 – 45 สร้างตัวแปร เก็บ ข้อมูลที่รับมาจากตัวแปร \$_POST

บรรทัดที่ 47 สร้างตัวแปร \$sql เก็บค่าคำสั่ง SQL สำหรับ UPDATE ข้อมูลที่มี id เท่ากับ ตัวแปร \$id

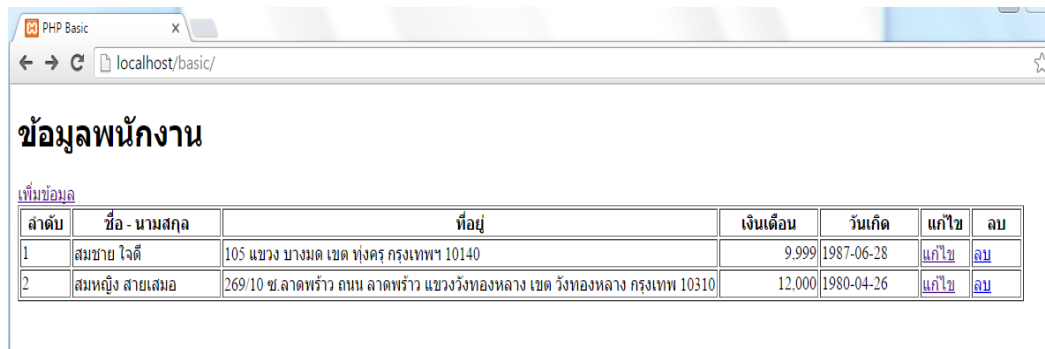
บรรทัดที่ 50 ใช้คำสั่ง if ตรวจสอบว่า ฟังก์ชัน mysqli_query() มีการทำงาน และส่งผลกลับมาเป็น จริงหรือไม่

บรรทัดที่ 52 – 54 แสดงผลข้อความหากมีการทำงานเป็นจริง

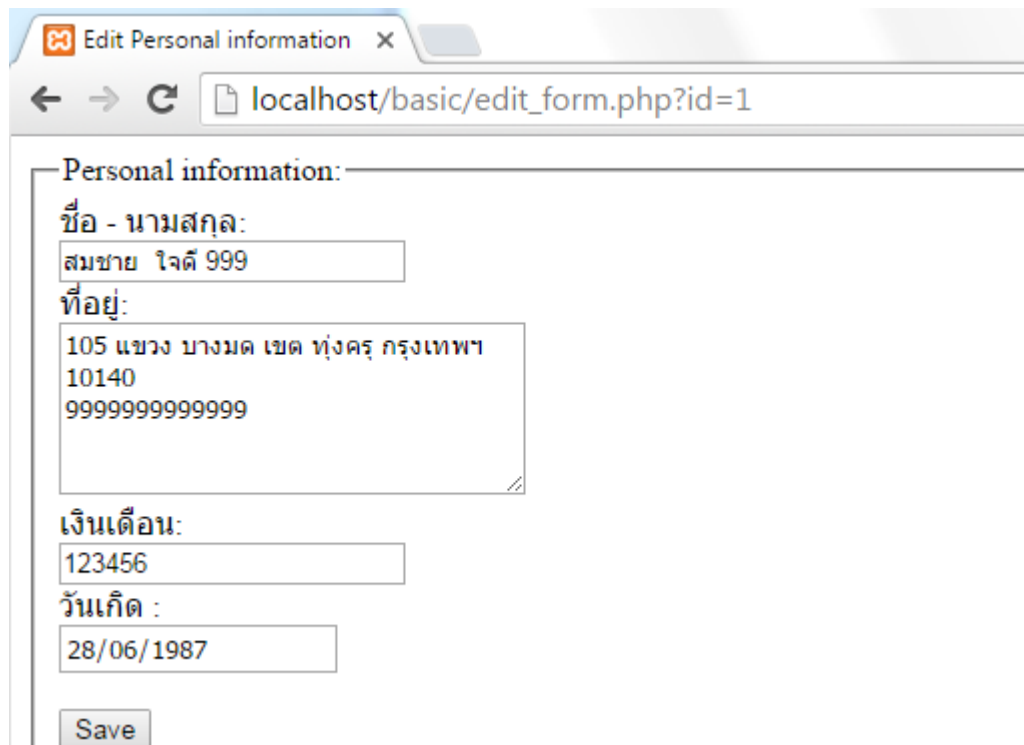
บรรทัดที่ 57 แสดงผล Error กรณีมีการทำงานเป็น เท็จ หรือ มีข้อผิดพลาด

ทดสอบการทำงาน

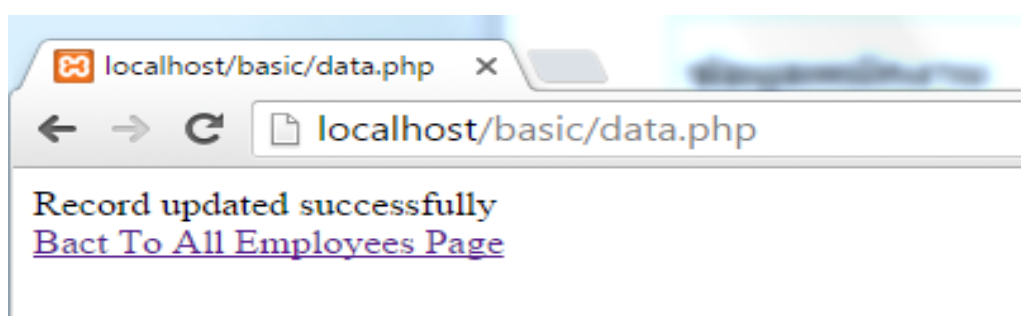
- คลิกปุ่มแก้ไข ในหน้า index.php



- เพิ่มข้อมูลทดสอบการแก้ไข



- เมื่อแก้ไขข้อมูลสำเร็จ



การลบข้อมูล

ในปุ่ม ลบข้อมูล จะมีการส่งค่าตัวแปรในรูปแบบ GET มีชื่อว่า delete_id มีค่าเท่ากับ id ของข้อมูล และใช้ attribute onclick เรียกใช้ ฟังก์ชัน confirm เพื่อแสดง ปุ่มยืนยันการลบข้อมูล

ตัวอย่างไฟล์ index.php ส่วนของการลบข้อมูล

```
echo"<td><a href='data.php?delete_id=".$row['id']."'
onclick='return confirm(\"คุณต้องการลบข้อมูลหรือไม่\")'>ลบ</a>
</td>";
```

เพิ่มเติม SourceCode ในไฟล์ data.php สำหรับ ลบ ข้อมูล

```
64 //delete data
65 ▼ if(isset($_GET['delete_id'])){
66
67     $id = $_GET['delete_id'];
68
69     // sql to delete a record
70     $sql = "DELETE FROM employees WHERE id= $id";
71
72 ▼ if (mysqli_query($conn, $sql)) {
73
74     echo "Record deleted successfully";
75
76     echo "<br><a href='index.php'>Bact To All Employees Page</a>";
77
78 ▼ } else {
79
80     echo "Error deleting record: " . mysqli_error($conn);
81
82
83 }
84
85 ?>
```

บรรทัดที่ 65 ใช้คำสั่ง if และ คำสั่ง isset ตรวจสอบ ว่ามีการส่งค่าตัวแปร \$_GET['delete_id'] มาหรือไม่

บรรทัดที่ 67 สร้างตัวแปร \$id กำหนดให้มีค่าเท่ากับ \$_GET['delete_id']

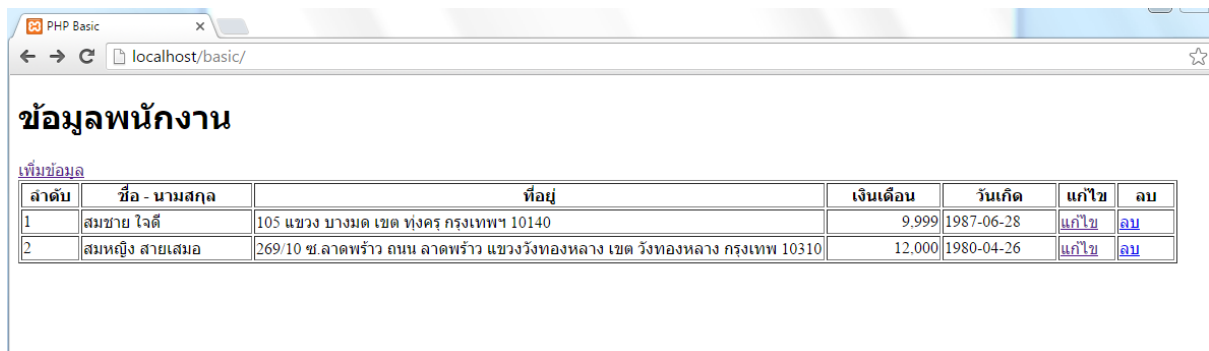
บรรทัดที่ 70 สร้างตัวแปร \$sql เก็บคำสั่ง SQL ลบข้อมูลที่มี id เท่ากับ ตัวแปร \$id

บรรทัดที่ 72 ใช้คำสั่ง if ตรวจสอบการทำงานของฟังก์ชัน mysqli_query() ว่าเป็นค่าจริง หรือไม่

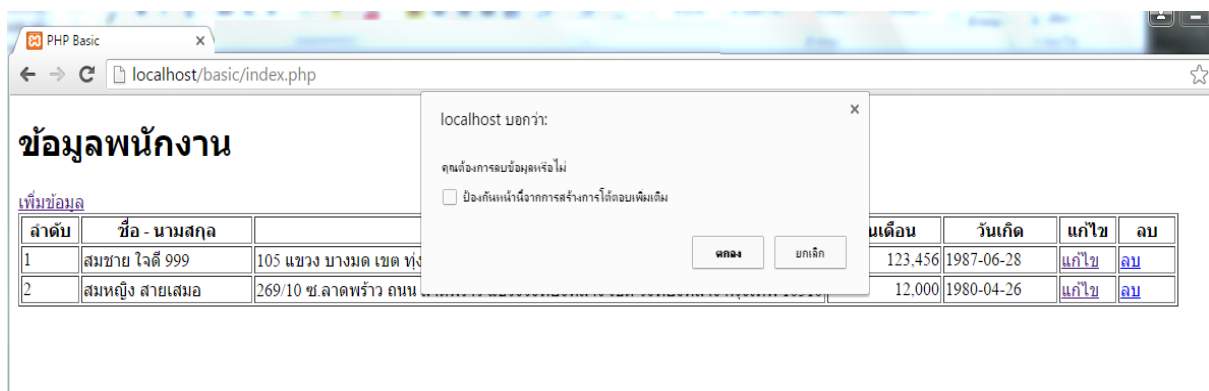
บรรทัดที่ 74 – 75 แสดงผลกรณีคำสั่ง if มีการทำงานเป็นจริง

บรรทัดที่ 79 แสดงผล Error กรณีมีค่าเป็น เท็จ หรือ มี Error เกิดขึ้น

ทดสอบการลบข้อมูล



เมื่อคลิกที่ ปุ่มลบ ข้อมูล จะมี ปุ่มยืนยันการลบข้อมูล แสดงขึ้นมา



เมื่อ คลิกปุ่ม ตกลง

