

The background features a series of colorful rays (green, orange, blue, grey) emanating from a central point on the left. The right side has a light grey grid with faint binary code (0s and 1s) and a line graph with two fluctuating lines, one orange and one blue.

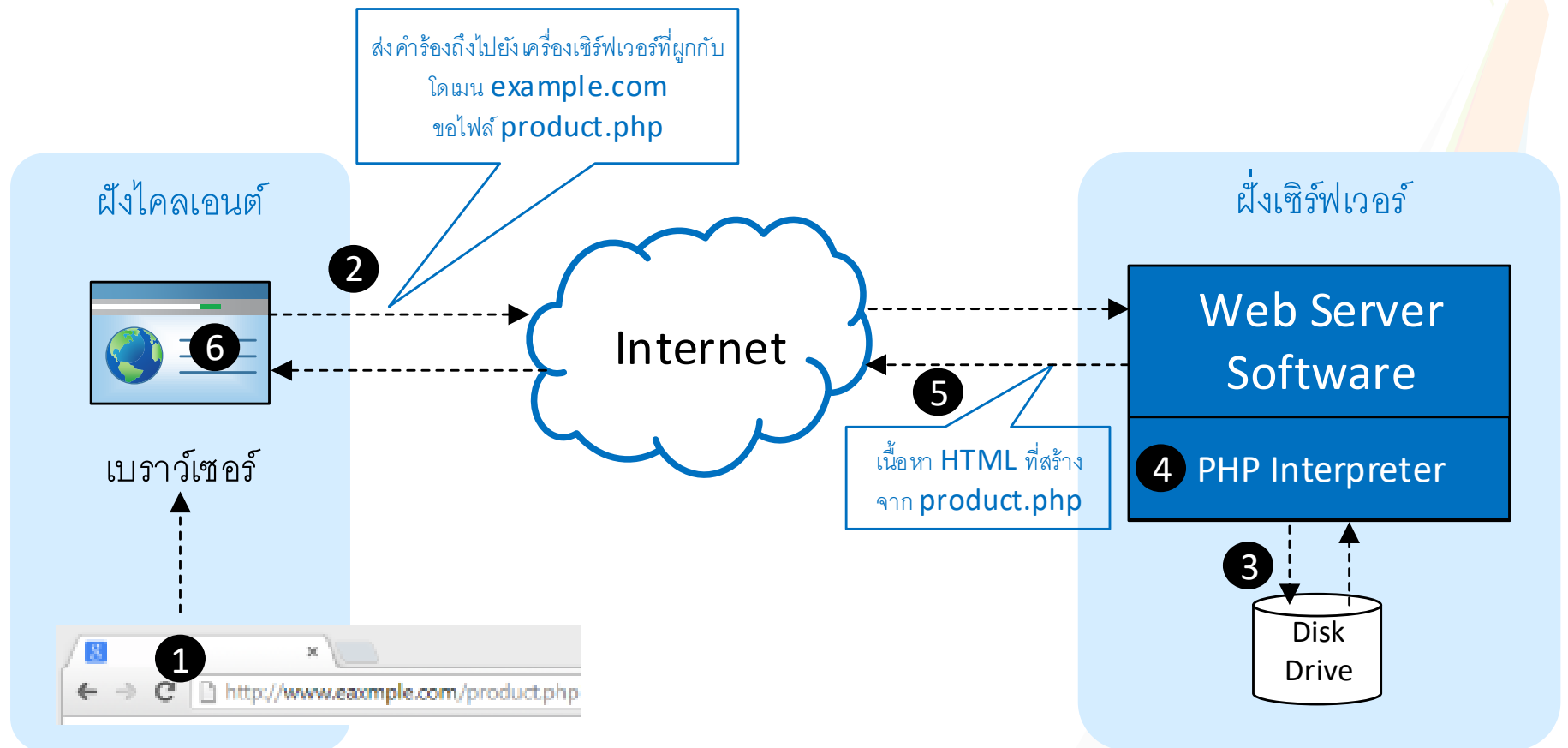
Chapter 6

PHP

(**P**HP **H**ypertext **P**reprocessor)

Theerayut Thongkrau

➡ สถาปัตยกรรม Web Application



➡ ทำไมต้องใช้ภาษาที่ทำงานบน Server

- เพื่อใช้รับข้อมูลจาก browser ผ่าน URL หรือแบบฟอร์ม HTML มาประมวลผลกับฐานข้อมูล
 - แสดงและค้นหาข้อมูล
 - เพิ่ม/ลบ/แก้ไข ข้อมูล
- เพื่อจัดการ Session และ Cookie
- เพื่อการติดต่อสื่อสารแบบ Asynchronous

➡ การแทรกโค้ด PHP บนหน้าเว็บ

- ภาษา PHP จะแทรกที่ส่วนใดของหน้าเว็บก็ได้ มี 2 แบบ
- แบบที่ 1 แบบเต็ม

<?php

...

?>

- แบบที่ 2 แบบย่อ (ไม่แนะนำ)

<?

...

?>

➡ คำสั่งในการส่งผลลัพธ์ไปยัง client

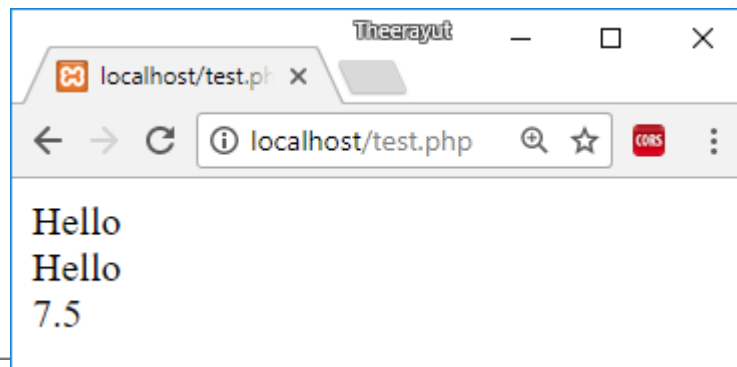
- คำสั่ง echo ใช้ในการกำหนดรูปแบบผลลัพธ์กลับไปยัง client สามารถเขียนได้ 2 แบบ

แบบเต็ม → `<?php echo "Hello"; ?>
`

แบบย่อ {
`<?= "Hello" ?>
`
`<?= 15/2 ?>`

สามารถปิดแท็ก และเปิดแท็ก
`<?php ?>` ใหม่กี่ครั้งก็ได้

การปิดแท็ก php ช่วยให้สามารถ
แสดงผลคำสั่ง HTML หรือข้อความ
บนหน้าเว็บโดยไม่ต้องใช้คำสั่ง echo



➡ การแปดโค้ด PHP บน Server

ฝั่งเซิร์ฟเวอร์

```
<html>
```

```
<body>
```

```
<?php
```

```
    echo "Hello World";
```

```
?>
```

```
</body>
```

```
</html>
```

โค้ด PHP บน Server

ฝั่งไคลเอนต์

```
<html>
```

```
<body>
```

Hello World

```
</body>
```

```
</html>
```

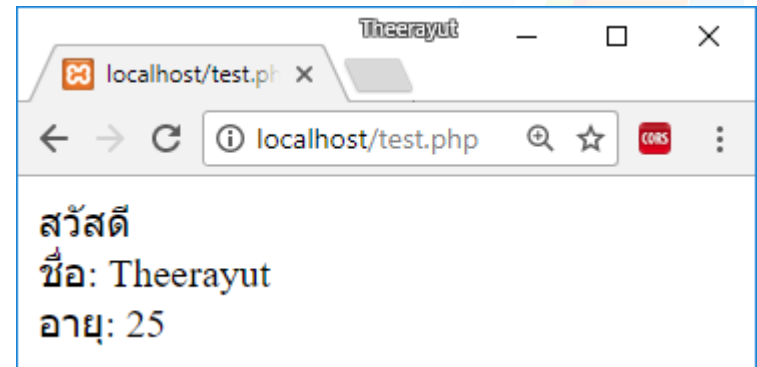
โค้ดที่ได้หลังจาก PHP ประมวลผลเสร็จแล้ว

➡ ตัวแปร

- PHP ไม่จำเป็นต้องประกาศตัวแปร และระบุชนิดข้อมูล
- การกำหนดค่าให้กับตัวแปรครั้งแรกถือเป็นการประกาศตัวแปร
- ตัวแปรใน PHP จะใช้เครื่องหมาย \$ แล้วตามด้วยชื่อของตัวแปร
- การกำหนดค่าให้ตัวแปร String ใช้ "..." หรือ '...' ก็ได้ ส่วนข้อมูลที่เป็นตัวเลขไม่ต้องมีสัญลักษณ์

```
<?php
$message = "สวัสดี<br>";
echo $message;
$myName = 'Theerayut';
$age = 25;
```

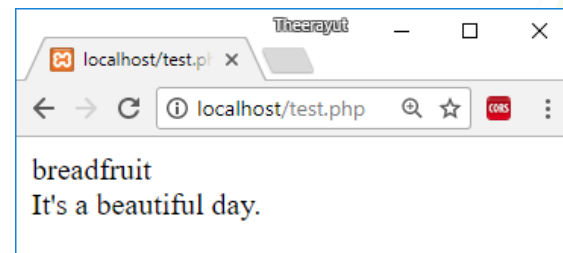
```
?>
ชื่อ: <?=$myName?><br>
อายุ: <?=$age?>
```



➡ การต่อ String

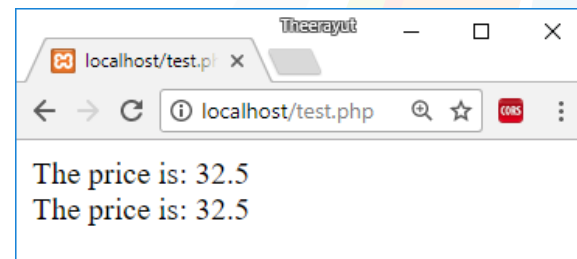
- การต่อ String จะใช้เครื่องหมายจุด (ไม่ใช่ +) เช่น

```
<?php
    echo 'bread' . 'fruit<br>';
    echo "It's a beautiful " . 'day.';
?>
```



- การแสดงค่าของตัวแปรภายใน String (ใช้กับอาร์เรย์ไม่ได้)

```
<?php
    $price = 32.5;
    echo "The price is: $price<br>";
    echo "The price is: " . $price;
?>
```



➡ ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	คำอธิบาย	ตัวอย่าง
+	การบวก	$\$x + \y
-	การลบ	$\$x - \y
*	การคูณ	$\$x * \y
/	การหาร	$\$x / \y
%	การหารเอาเศษ (Modulo)	$\$x \% \y

<?php

```
$x = 10;
```

```
$y = 6;
```

```
echo $x + $y . "<br>;"
```

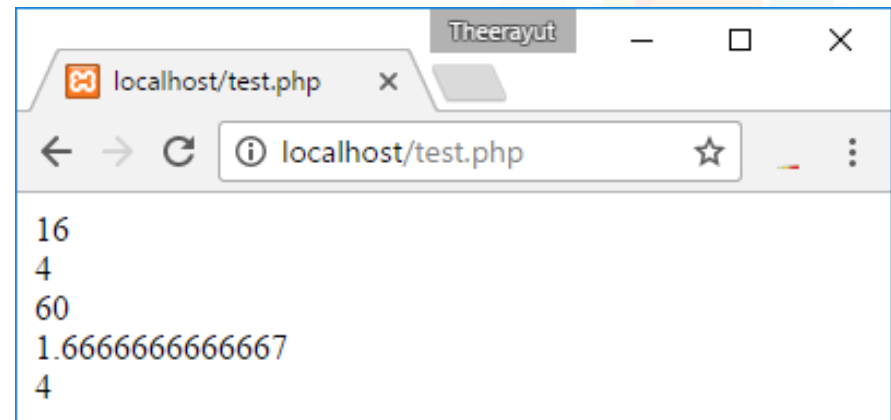
```
echo $x - $y . "<br>;"
```

```
echo $x * $y . "<br>;"
```

```
echo $x / $y . "<br>;"
```

```
echo $x % $y . "<br>;"
```

?>



➡ ตัวดำเนินการเปรียบเทียบ

Operator	คำอธิบาย	ตัวอย่าง
==	เท่ากัน	$x == y$
===	เท่ากันทั้งค่า และชนิดข้อมูล	$x === y$
!= หรือ <>	ไม่เท่ากัน	$x != y$ หรือ $x <> y$
!==	ไม่เท่ากัน ค่าต่างกัน หรือ ชนิดข้อมูลต่างกัน	$x !== y$
>	มากกว่า	$x > y$
<	น้อยกว่า	$x < y$
>=	มากกว่าเท่ากับ	$x >= y$
<=	น้อยกว่าเท่ากับ	$x <= y$

➡ ตัวดำเนินการตรรกะ

ตัวดำเนินการ	ชื่อ	ตัวอย่าง
and หรือ &&	และ	\$x and \$y
or หรือ	หรือ	\$x or \$y
!	ไม่	!\$x
xor	Xor	\$x xor \$y

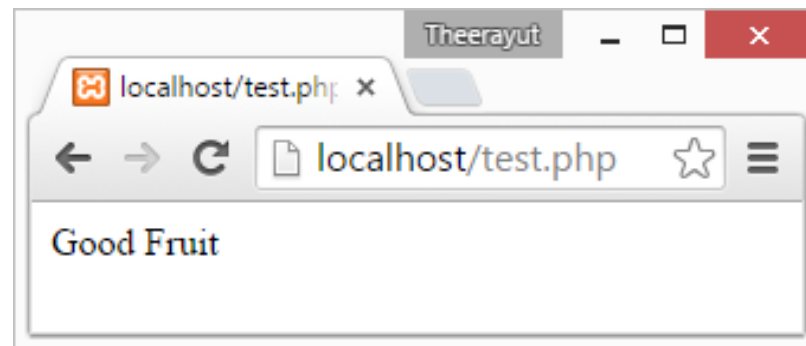
➡ คำสั่งเงื่อนไข if

แบบที่ 1

```
<html>
<body>
<?php
    $fruit = "apple";
    if ($fruit == "apple") {
        echo "Good Fruit";
    }
?>
</body>
</html>
```

แบบที่ 2

```
<html>
<body>
<?php $fruit = "apple"; ?>
<?php if ($fruit == "apple"): ?>
    Good Fruit
<?php endif; ?>
</body>
</html>
```



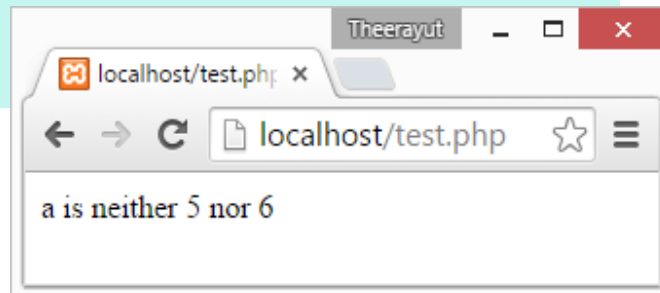
➡ คำสั่งเงื่อนไข if...else

แบบที่ 1

```
<html>
<body>
<?php
    $a = 9;
    if ($a == 5) {
        echo "a equals 5";
    } else if ($a == 6) {
        echo "a equals 6";
    } else {
        echo "a is neither 5 nor 6";
    }
?>
</body>
</html>
```

แบบที่ 2

```
<html>
<body>
<?php
    $a = 9;
    if ($a == 5):
        ?>
        a equals 5
    <?php elseif ($a == 6): ?>
        a equals 6
    <?php else: ?>
        a is neither 5 nor 6
    <?php endif; ?>
</body>
</html>
```



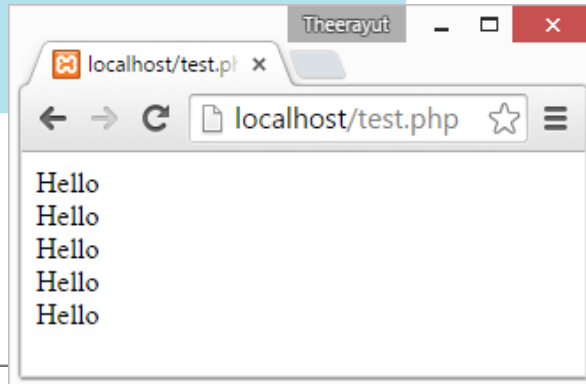
➡ คำสั่งวนซ้ำ while

แบบที่ 1

```
<html>
<body>
<?php
    $i = 1;
    while ($i <= 5) {
        echo "Hello<br>";
        $i++;
    }
?>
</body>
</html>
```

แบบที่ 2

```
<html>
<body>
<?php
    $i = 1;
    while ($i <= 5) :
?>
    Hello<br>
<?php
    $i++;
endwhile;
?>
</body>
</html>
```

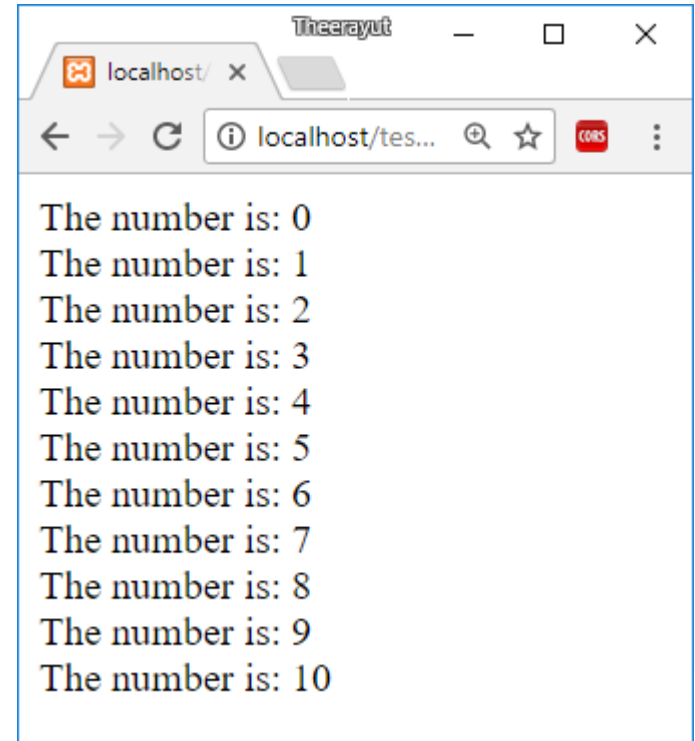


➡ คำสั่งวนซ้ำ for

<?php

```
for ($x=0; $x<=10; $x++) {  
    echo "The number is: $x <br>";  
}
```

?>



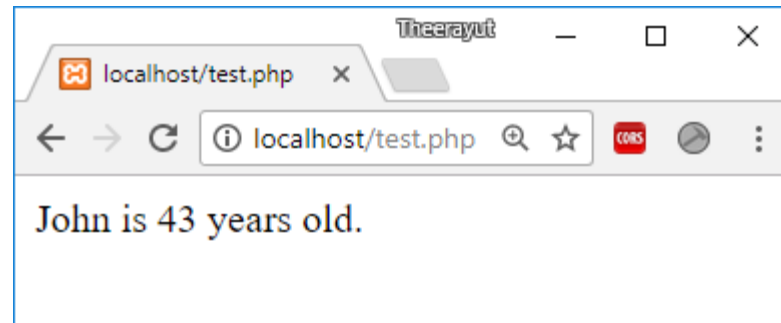
➔ Associative Arrays

- Associative Arrays คือ อาร์เรย์ที่ใช้ชื่อในการระบุตำแหน่งช่องอาร์เรย์

```
<?php
    $person = array("fullname"=>"John", "height"=>"173", "age"=>"43");
    echo $person["fullname"] . " is " . $person["age"] . " years old.<br>";
?>
```

ชื่อ index

ค่า/ข้อมูล



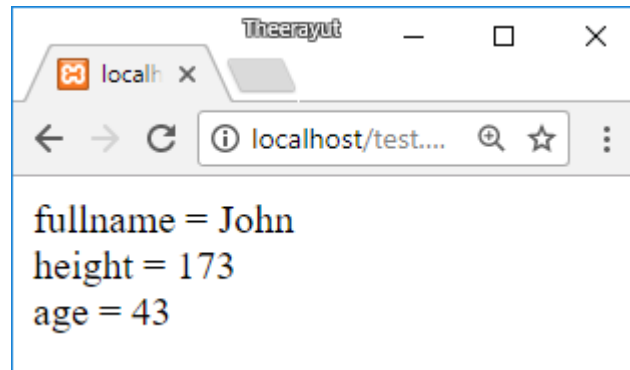
➡ การใช้ foreach กับ Associative Arrays

```
foreach (ตัวแปรอาร์เรย์ as ตัวแปรเก็บค่า index => ตัวแปรเก็บค่าสมาชิก) {  
    คำสั่ง;  
}
```

```
<?php
```

```
$person = array("fullname"=>"John", "height"=>"173", "age"=>"43");
```

```
foreach ($person as $key => $value) {  
    echo $key . " = " . $value . "<br>";  
}  
?>
```

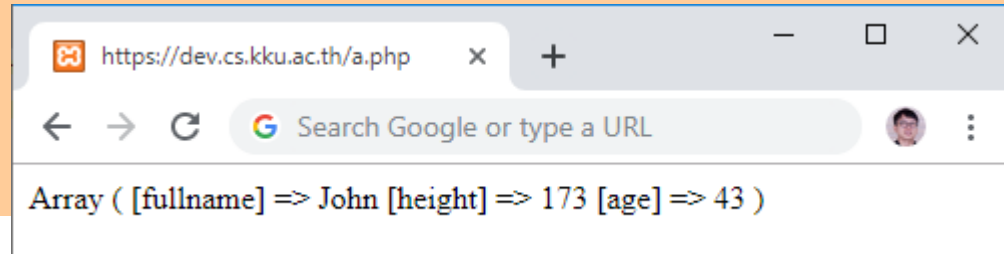


➡ ฟังก์ชันแสดงค่าทั้งหมดในอาร์เรย์

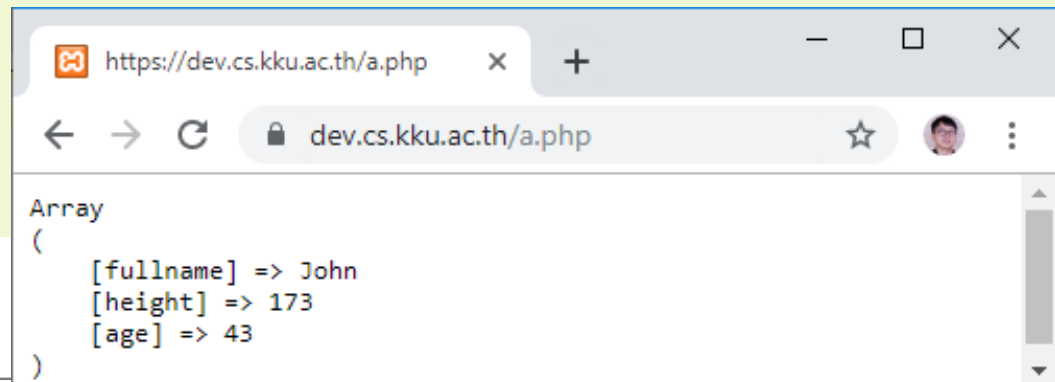
- หากต้องการดูค่าทั้งหมดในอาร์เรย์สามารถใช้ฟังก์ชัน `print_r()`;

```
<?php
$person = array("fullname"=>"John", "height"=>"173", "age"=>"43");

// แสดงค่าทั้งหมดในอาร์เรย์
print_r($person);
?>
```



```
<?php
$person = array("fullname"=>"John", "height"=>"173", "age"=>"43");
// ใช้แท็ก <pre> ช่วยให้แสดงผลได้เข้าใจมากขึ้น
echo "<pre>";
print_r($person);
echo "</pre>";
?>
```



➡ การส่งข้อมูลไปยัง Server

- **แบบ GET**

- ส่งโดยแนบข้อมูลต่อท้าย URL
- ส่งโดยสร้างฟอร์ม HTML และระบุ method="get"

- **แบบ POST**

- ส่งโดยสร้างฟอร์ม HTML และระบุ method="post"

➡ การส่งข้อมูลแบบ GET ผ่าน URL

- การส่งข้อมูลไปยัง server โดยแนบไปกับ URL จะขึ้นต้นด้วยเครื่องหมาย ?
- สามารถกำหนดชื่อและค่าในรูปแบบ name=value
- คั่นแต่ละข้อมูลด้วยเครื่องหมาย &
- ข้อมูลที่ส่งไปยัง server เรียกว่า **Request Parameter**

http://dev.cs.kku.ac.th/profileList.php?major=it&year=2560§ion=1

The diagram illustrates the components of the URL `http://dev.cs.kku.ac.th/profileList.php?major=it&year=2560§ion=1`. Annotations include:

- ชื่อ Domain** (Domain Name): Points to `dev.cs.kku.ac.th`.
- ชื่อไฟล์ php ที่รับ request parameter** (PHP file name receiving request parameters): Points to `profileList.php`.
- จุดเริ่มต้นของ Request Parameter** (Start of Request Parameter): Points to the question mark `?`.
- ข้อมูลแรก** (First data): Points to `major=it`.
- ข้อมูลที่ 2** (Second data): Points to `year=2560`.
- ข้อมูลที่ 3** (Third data): Points to `section=1`.
- มีหลาย parameter ขึ้นด้วย &** (Multiple parameters separated by &): Points to the ampersand symbols separating the parameters.

➡ อารเรย์ \$_GET

- เมื่อ server รับข้อมูลแล้ว PHP Interpreter จะสร้างตัวแปรอาร์เรย์ชนิด Associative ชื่อ \$_GET อัตโนมัติเพื่อเก็บข้อมูลจาก URL

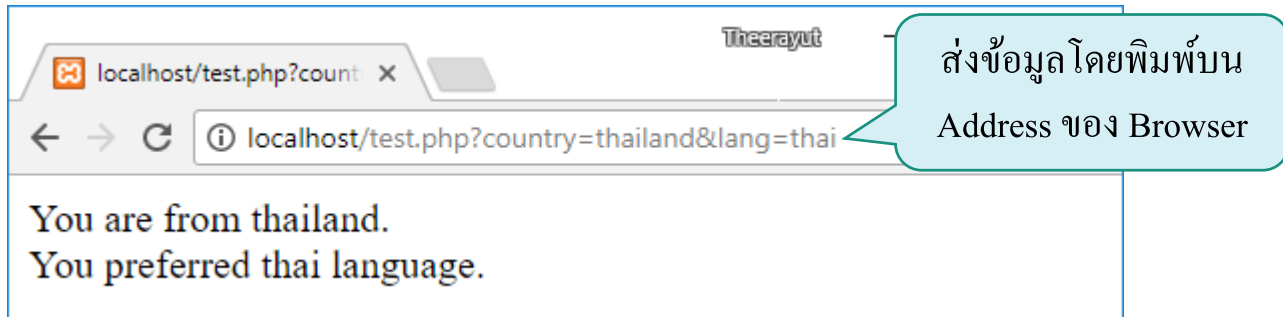
ตัวอย่าง URL

http://localhost/test.php?country=thailand&lang=thai

Array ที่ถูกสร้างอัตโนมัติ

`$_GET = array("country"=>"thailand", "lang"=>"thai");`

➡ ตัวอย่างการส่งและรับข้อมูลแบบ GET



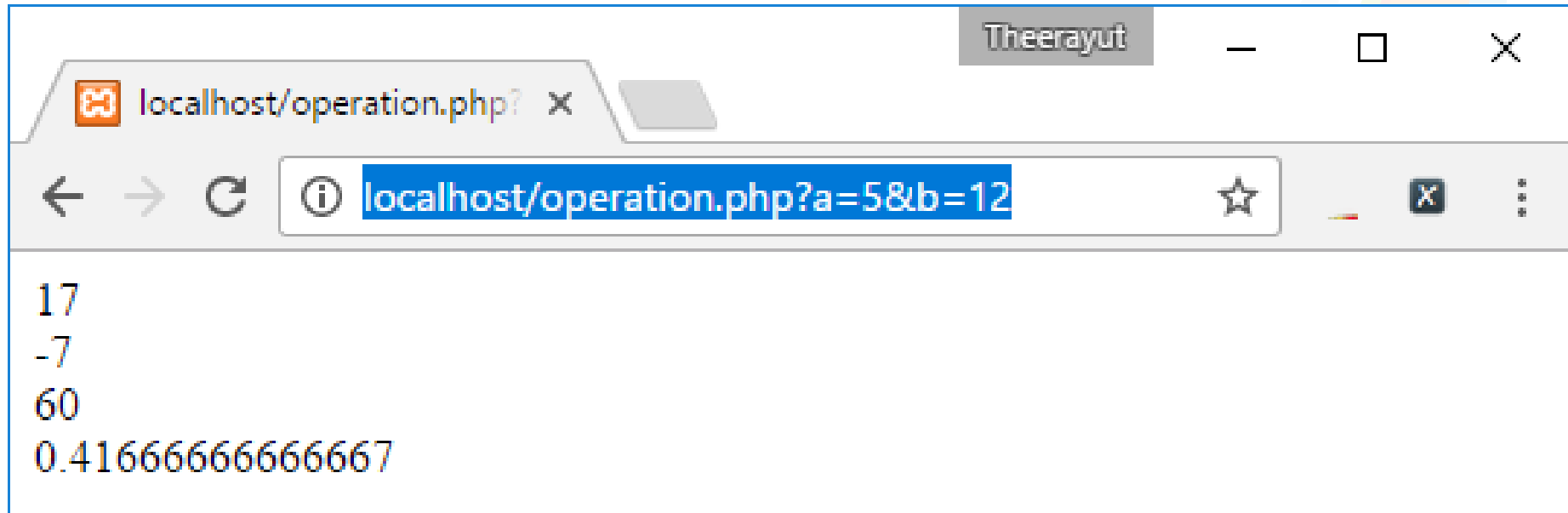
ไฟล์ test.php

```
<html>
<body>
<?php
    $country = $_GET["country"];
    $language = $_GET["lang"];
    echo "You are from " . $country . "<br>";
    echo "You preferred " . $language . " language.";
?>
</body>
</html>
```

เมื่อ Server รับข้อมูลแล้ว จะสร้างอาร์เรย์ `$_GET` อัตโนมัติ นักพัฒนาสามารถอ้างอิงค่าจากสมาชิกตามรูปแบบของ Associative Array ได้ทันที

➡ กิจกรรม

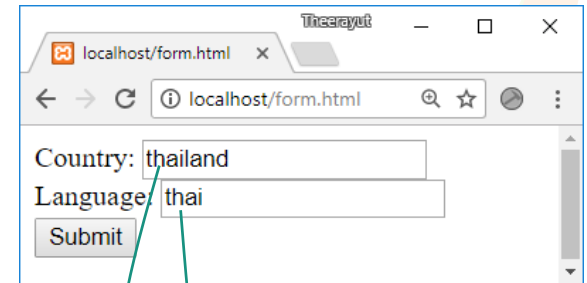
จงเขียน โค้ด PHP ตั้งชื่อไฟล์ operation.php เพื่อรับค่า a และ b จาก url เพื่อนำค่ามาบวก ลบ คูณ และหาร หลังจากนั้นทดสอบส่งข้อมูลแบบ GET



➡ การส่งข้อมูลแบบ GET ผ่านฟอร์ม HTML

ไฟล์ form.html

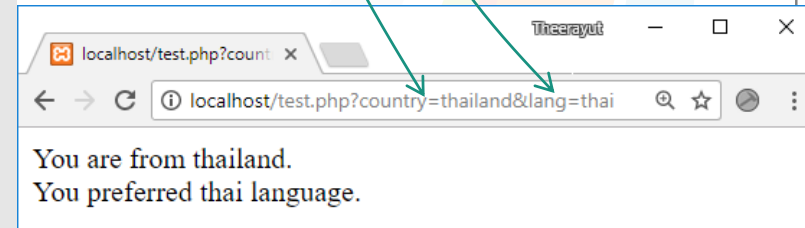
```
<html>
<body>
<form action="test.php" method="get">
Country: <input type="text" name="country"><br>
Language: <input type="text" name="lang"><br>
<input type="submit">
</form>
```



เมื่อกดปุ่ม submit จะได้ส่งข้อมูล
ไปยัง test.php

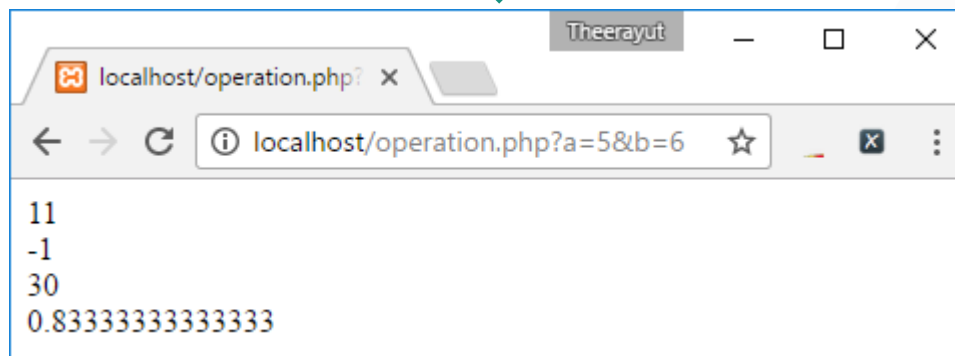
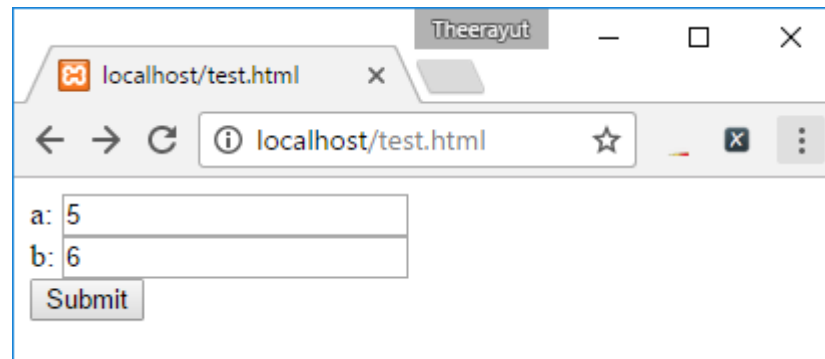
ไฟล์ test.php

```
<html>
<body>
<?php
    $country = $_GET["country"];
    $language = $_GET["lang"];
    echo "You are from " . $country . "<br>";
    echo "You preferred " . $language . " language.";
?>
</body>
</html>
```



➡ กิจกรรม

จงสร้างแบบฟอร์ม HTML สำหรับส่งค่า a และ b ไปยังไฟล์ PHP ชื่อ operation.php แทนการพิมพ์จาก URL โดยตรง

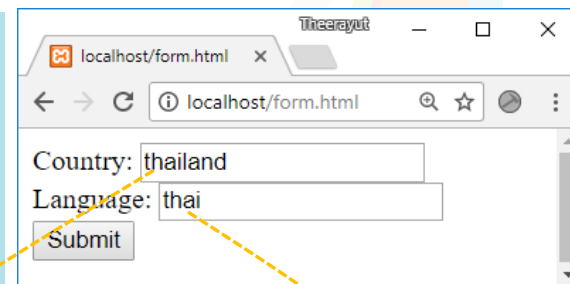


➡ การรับข้อมูลแบบ POST

- PHP Interpreter จะสร้างตัวแปรอาร์เรย์ชนิด Associative ชื่อ `$_POST` แบบอัตโนมัติเพื่อเก็บข้อมูลที่ส่งมาแบบเบื้องหลัง

ตัวอย่างฟอร์ม

```
<form action="test.php" method="post">  
  Country: <input type="text" name="country"><br>  
  Language: <input type="text" name="lang"><br>  
  <input type="submit">  
</form>
```



The screenshot shows a web browser window with the address bar displaying 'localhost/form.html'. The page content includes a form with two text input fields: 'Country:' containing 'thailand' and 'Language:' containing 'thai'. Below these fields is a 'Submit' button. Red and yellow arrows from the code blocks point to the 'country' and 'lang' input fields respectively.

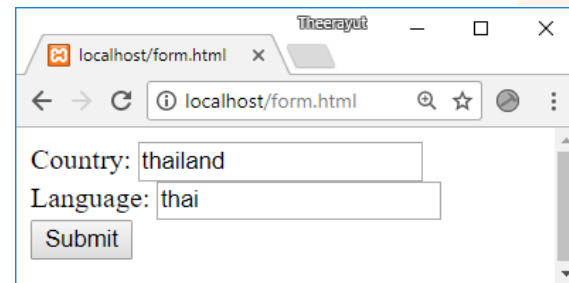
Array ที่ถูกสร้างอัตโนมัติ

```
$_POST = array("country"=>"thailand", "lang"=>"thai");
```

➡ การส่งข้อมูลแบบ POST ผ่านฟอร์ม HTML

ไฟล์ form.html

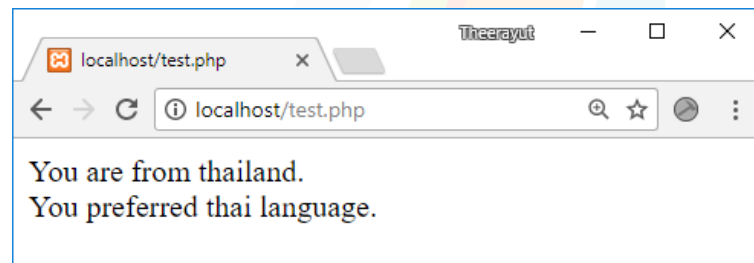
```
<html>
<body>
<form action="test.php" method="post">
  Country: <input type="text" name="country"><br>
  Language: <input type="text" name="lang"><br>
  <input type="submit">
</form>
```



ไฟล์ test.php

```
<html>
<body>
<?php
  $country = $_POST["country"];
  $language = $_POST["lang"];
  echo "You are from " . $country . "<br>";
  echo "You preferred " . $language . " language.";
?>
</body>
</html>
```

เมื่อกดปุ่ม submit จะได้ส่งข้อมูล
ไปยัง test.php โดยจะไม่เห็นข้อมูลทาง URL



➡ เปรียบเทียบวิธีการส่งข้อมูลบน internet

แบบ GET	แบบ POST
ส่งข้อมูลผ่าน URL ประกอบด้วย คู่ของชื่อตัวแปรและค่า	ส่งข้อมูลแบบเบื้องหลัง
หากฟอร์มไม่ระบุว่าเป็น <code>method="?"</code> ในแท็ก <code><form></code> จะส่งแบบ GET เป็นค่า default	หากต้องการส่งแบบนี้ต้องระบุ attribute <code>method="post"</code> ในแท็ก <code><form></code>
ผู้ใช้สามารถเห็นข้อมูลบน URL ของเบราว์เซอร์ได้	ผู้ใช้ไม่สามารถมองเห็นข้อมูลบน URL ของเบราว์เซอร์ได้
สามารถเก็บ Bookmark ไว้ได้ และส่งข้อมูลได้อีกเพียงเรียก URL เดิม	ไม่สามารถเก็บ Bookmark ไว้ได้ การส่งข้อมูลซ้ำต้องกรอกข้อมูลใหม่
จำกัดขนาดข้อมูลที่ 2000 อักขระ	ไม่จำกัดขนาดข้อมูล
อ่านข้อมูลจากตัวแปร <code>\$_GET</code> หรือ <code>\$_REQUEST</code>	อ่านข้อมูลจากตัวแปร <code>\$_POST</code> หรือ <code>\$_REQUEST</code>