มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

บทที่ 4 ภาษา PHP

รายวิชา 3603610 การพัฒนาเว็บแอปพลิเคชันสำหรับสตาร์ทอัพ

ผศ.ดร.เดช ธรรมศิริ

PHP คืออะไร?

- PHP เป็นตัวย่อของ "PHP: Hypertext Preprocessor"
- PHP เป็นภาษาสคริปต์โอเพ่นซอร์สที่ใช้กันอย่างแพร่หลาย
- สคริปต์ PHP จะถูกดำเนินการบนเซิร์ฟเวอร์
- PHP ดาวน์โหลดและใช้งานได้ฟรี

PHP สามารถทำอะไรได้บ้าง?

- PHP สามารถสร้างเนื้อหาหน้าแบบไดนามิกได้
- PHP สามารถสร้าง เปิด อ่าน เขียน ลบ และปิดไฟล์บนเซิร์ฟเวอร์ได้
- PHP สามารถรวบรวมข้อมูลแบบฟอร์มได้
- PHP สามารถส่งและรับคุกกี้ได้
- PHP สามารถเพิ่ม ลบ แก้ไขข้อมูลในฐานข้อมูลของคุณได้
- PHP สามารถใช้เพื่อควบคุมการเข้าถึงของผู้ใช้
- PHP สามารถเข้ารหัสข้อมูลได้

ไวยากรณ์ PHP ขั้นพื้นฐาน

คุณสามารถวางสคริปต์ PHP ไว้ที่ใดก็ได้ในเอกสาร

สคริปต์ PHP เริ่มต้น <?phpและลงท้ายด้วย ?>:

```
<?php
// PHP code goes here
?>
```

นามสกุลไฟล์เริ่มต้นสำหรับไฟล์ PHP คือ " .php"

ไฟล์ PHP โดยทั่วไปมีแท็ก HTML และโค้ดสคริปต์ PHP บางส่วน

ตัวอย่าง ไฟล์ .php ที่มีทั้งโค้ด HTML และโค้ด PHP:

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

รูปแบบการแสดงความคิดเห็นในโค้ด PHP:

- ความคิดเห็นบรรทัดเดียวเริ่มต้น//ด้วย
- ข้อความใด ๆ ระหว่าง//และ สิ้นสุดบรรทัดจะถูกละเว้น (จะไม่ถูกดำเนินการ)
- คุณสามารถใช้#สำหรับความคิดเห็นบรรทัดเดียวได้ แต่ในบทช่วยสอนนี้เราจะใช้//.
- ตัวอย่างต่อไปนี้ใช้คำอธิบายบรรทัดเดียว:
- // Outputs a welcome message:
- echo "Welcome Home!";

ตัวแปร PHP

ขึ้นต้นด้วย $
ตามด้วย A-Z หรือ a-z หรือ _
ตามด้วย A-Z หรือ a-z หรือ 0-9 หรือ _ เช่น $myvar; $my_var; $myVar;
Case Sensitive ตัวพิมพ์ใหญ่/เล็กถือเป็นคนละตัว เช่น $myvar; $Myvar; $MyVar; $myVar;
ไม่ตั้งชื่อซ้ำคำสงวน เช่น $_POST; $_SESSION; $_GET;

ชนิดของตัวแปร

- Boolean -> True , False
- Integer -> เลขจำนวนเต็ม
- Float -> เลขจำนวนจริง
- String -> ตัวอักษรที่นำไปคำนวณทางคณิตศาสตร์ไม่ได้
- Array -> ตัวแปรชุด
- Object -> เก็บคุณสมบัติของ Object
- Resource -> สำหรับอ้างอิงถึงแหล่งภายนอก เช่น การเปิดไฟล์ข้อมูล การเชื่อมต่อฐานข้อมูล
- Null -> ตัวแปรที่ไม่มีค่าอะไรเลยเรียกว่ามีค่าเป็น Null เช่น เมื่อประกาศตัวแปรแล้วแต่ยังไม่ได้กำหนดค่าใดๆให้ตัวแปร กำหนดค่าให้ตัวแปรมีค่าเป็น Null
  $MySalary = NULL;

การประกาศตัวแปรใน PHP

- $txt = "ข้อความ";
- $x = 5;
- $y = 10.5;

- ตัวแปรการแสดงผล
- ตัวอย่างต่อไปนี้จะแสดงวิธีการแสดงข้อความและตัวแปรด้วย printคำสั่ง:

```
$txt1 = "Learn PHP";
$txt2 = “Dech  Thammasiri";
print "<h2>$txt1</h2>";
print "<p>สวัสดีครับ ผมชื่อ $txt2</p>";
```

การแสดงผลด้วยคำสั่ง echo

- Echo เป็นคำสั่งในการแสดงผลข้อความออกทางหน้าจอ echo ไม่ใช่ฟังก์ชันเสียทีเดียวแต่มันเป็นโครงสร้างของภาษา PHP มาดูตัวอย่างการใช้งานในแบบต่างๆ

```
<?php

echo "Bound to pass the point of contemplation.\n";
echo "Mercury ", "Venus ", "Earth ", "Mars", "\n";

$lang = "PHP";
echo "$lang is a language for web.\n";

$a = 10;
$b = 5;
echo "a + b = ", ($a + $b);

?>
```

ตัวดำเนินการ (Assignment operator)

- ตัวดำเนินการกำหนดค่า (Assignment operator) คือตัวดำเนินการที่ใช้สำหรับกำหนดค่าให้กับตัวแปรหรือค่าคงที่ เครื่องหมายเท่ากับ = ใช้เป็นสัญลักษณ์ของตัวดำเนินการนี้ การทำงานของตัวดำเนินการนั้นจะนำค่าจากด้านขวามาใส่ตัวแปรทางด้านซ้าย

- $x = 1;
- $y = $x + 5;
- $text = "PHP language";

- ในตัวอย่างเป็นการใช้งานตัวดำเนินการกำหนดค่าในการกำหนดค่าให้กับตัวแปรในภาษา PHP เราได้กำหนดค่า 1 ให้กับตัวแปร $x เรากำหนดค่าให้กับตัวแปร $y ในรูปแบบของ Expression และกำหนด String ให้กับตัวแปร $text

- นี่เป็นการดำเนินการที่พื้นฐานที่สุดในการเขียนโปรแกรม

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)

- เป็นตัวดำเนินการในการหาผลลัพธ์จากการกระทำทางคณิตศาสตร์ เช่น การบวก การลบ การคูณ และหารหาร โดยมีตัวเลขเป็น Operand และจะได้ผลลัพธ์สุดท้ายออกมาค่าเดียว

Symbol	Name	Example
+	บวก(Addition)	$c = $a + $b
-	ลบ(Subtraction)	$c = $a - $b
*	คูณ(Multiplication)	$c = $a * $b
/	หาร(Division)	$c = $a / $b
%	หารหาเศษ(Modulo)	$c = $a % $b

Compound assignment operators

- ตัวดำเนินการที่รวมระหว่างตัวดำเนินการทางคณิตศาสตร์และตัวดำเนินการกำหนดค่า มันใช้สำหรับอัพเดทข้อมูลที่อ้างอิงจากข้อมูลในปัจจุบัน หรือกล่าวอีกนัยหนึ่ง มันเป็นรูปแบบที่สั้นกว่าของตัวดำเนินการทางคณิตศาสตร์

Operator	Example	Equivalent to
+=	$a += 2;	$a = $a + 2
-=	$a -= 2;	$a = $a - 2
*=	$a *= 2;	$a = $a * 2
/=	$a /= 2;	$a= $a / 2
%=	$a %= 2;	$a = $a % 2
>>=	$a >>= 2;	$a = $a >> 2
<<=	$a <<= 2	$a = $a << 2
&=	$a & = 2;	$a = $a & 2
^=	$a ^= 2;	$a= $a ^ 2
|=	$a |= 2;	$a = $a | 2

ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

- ตัวดำเนินการที่ใช้สำหรับเปรียบเทียบระหว่างค่าสองค่า ผลลัพธ์ของการเปรียบเทียบจะได้เป็น Boolean ที่มีค่าเป็นเพียง True หรือ False เท่านั้น ตัวดำเนินการเปรียบเทียบมักจะใช้สำหรับคำสั่งเลือกเงื่อนไขหรือวนซ้ำ ในการเขียนโปรแกรม เราใช้ตัวดำเนินการเหล่านี้เพื่อสร้างเงื่อนไขให้โปรแกรมทำงานตามที่ต้องการ

- ในภาษา PHP เนื่องจากตัวแปรเป็นแบบ Typeless ดังนั้นค่าของตัวแปรอาจจะมีความหมายเท่ากันในบางบริบท เช่น ค่าของ True นั้นอาจจะเท่ากับ 1 และค่าของ False อาจจะเท่ากับ 0

ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

Operator	Example	Result
==	$a == $b	เป็นจริงถ้า a เท่ากับ b ไม่เช่นนั้นเป็นเท็จ
!=, <>	$a != $b	เป็นจริงถ้า a ไม่เท่ากับ b ไม่เช่นนั้นเป็นเท็จ
<	$a < $b	เป็นจริงถ้า a น้อยกว่า b ไม่เช่นนั้นเป็นเท็จ
<=	$a <= $b	เป็นจริงถ้า a น้อยกว่าหรือเท่ากับ b ไม่เช่นนั้นเป็นเท็จ
>	$a > $b	เป็นจริงถ้า a มากกว่า b ไม่เช่นนั้นเป็นเท็จ
>=	$a >= $b	เป็นจริงถ้า a มากกว่าหรือเท่ากับ b ไม่เช่นนั้นเป็นเท็จ
===	$a === $b	เป็นจริงถ้าทั้งค่าและประเภทของตัวแปร a เท่ากับ b ไม่เช่นนั้นเป็นเท็จ
!==	$a !== $b	เป็นจริงถ้าทั้งค่าและประเภทของตัวแปร a ไม่เท่ากับ b ไม่เช่นนั้นเป็นเท็จ

ตัวอย่างการใช้ตัวดำนำเนินการในการเปรียบเทียบค่าในตัวแปรในภาษา PHP


```php
<?php
$a = 5;
$b = 10;
if ($a == $b) {
    echo "a is equal to b\n";
} else {
    echo "a is not equal to b\n";
}
if ($a < $b) {
    echo "a is less than b\n";
} else {
    echo "a is not less than b\n";
}
if ($a % 2 == 0) {
    echo "a is even number\n";
} else {
    echo "a is odd number\n";
}
?>
```

ในตัวอย่าง เรามีตัวแปรสองตัวคือ $a และ $b และได้กำหนดค่าให้กับตัวแปรเหล่านี้ เราใช้ตัวดำเนินการเปรียบเทียบกับคำสั่งเงื่อนไข If เมื่อเงื่อนไขเป็นจริงคำสั่งในบล็อคของ If จะทำงาน และถ้าไม่เป็นจริงคำสั่งในบล็อคของ Else จะทำงาน

ในคำสั่ง if ($a == $b) เป็นการตรวจสอบถ้าหากตัวแปรมีค่าเท่ากัน เนื่องจากค่าในตัวแปรทั้งสองไม่เท่ากัน โปรแกรมจึงทำงานในบล็อคของ Else ในคำสั่ง if ($a < $b) เป็นการตรวจสอบถ้าหากตัวแปร $a มีค่าน้อยกว่า $b ซึ่งเป็นจริง และในคำสั่งสุดท้าย if ($a % 2 == 0) เป็นการตรวจสอบถ้าหาก $a เป็นเลขคู่จากหารเอาเศษ ซึ่งเป็นเท็จ

a is not equal to b
a is less than b
a is odd number
นี่เป็นผลลัพธ์เมื่อรันโปรแกรม

ตัวดำเนินการตรรกศาสตร์ (Logical operators)

- ตัวดำเนินการสำหรับสร้าง Expression จาก Expression ย่อยๆ โดยผลลัพธ์ที่ได้จะเป็น Boolean ที่มีค่า True หรือ False ต่อไปนี้เป็นตัวดำเนินการตรรกศาสตร์ในภาษา PHP

Operator	Example	Result
and	$expr1 and $expr2	เป็นจริงถ้าหาก Expression ทั้งสองเป็นจริง ไม่เช่นนั้นเป็นเท็จ
or	$expr1 or $expr2	เป็นจริงถ้าหากอย่างน้อยหนึ่ง Expression เป็นจริง ไม่เช่นนั้นเป็นเท็จ
xor	$expr1 xor $expr2	เป็นจริงถ้าหาก Expression ทั้งสองมีค่าที่ต่างกัน ไม่เช่นนั้นเป็นเท็จ
not	not $expr1	เป็นจริงถ้าหาก Expression เป็นเท็จ และเป็นเท็จถ้าหาก Expression เป็นจริง
&&	$expr1 && $expr2	เป็นจริงถ้าหาก Expression ทั้งสองเป็นจริง ไม่เช่นนั้นเป็นเท็จ (อีกรูปแบบของ and)
‖	$expr1 ‖ $expr2	เป็นจริงถ้าหากอย่างน้อยหนึ่ง Expression เป็นจริง ไม่เช่นนั้นเป็นเท็จ (อีกรูปแบบของ or)
!	!$expr1	เป็นจริงถ้าหาก Expression เป็นเท็จ และเป็นเท็จถ้าหาก Expression เป็นจริง (อีกรูปแบบของ not)

คำสั่งเลือกเงื่อนไข

- ในบทนี้ คุณจะได้เรียนรู้เกี่ยวกับคำสั่งเลือกเงื่อนไข If If Else และ Switch เพื่อควบคุมการทำงานของโปรแกรมในภาษา PHP

- ในการเขียนโปรแกรม อาจจะมีเงื่อนไขหรือข้อกำหนดบางอย่างที่คุณต้องการให้โปรแกรมทำงานแตกต่างกันไป การตัดสินใจจึงเป็นเรื่องธรรมดาที่เกิดขึ้นทั้งในการเขียนโปรแกรมและในชีวิตประจำวัน ยกตัวอย่างเช่น ถ้าคุณมีเงินมากกว่า 100 เหรียญ คุณจะซื้อวิดีโอเกม แต่ถ้าคุณไม่เงินไม่พอคุณจะซื้อหนังสือแทน

คำสั่ง If

- คำสั่ง If เป็นคำสั่งควบคุมที่พื้นฐานที่สุดในการเขียนโปรแกรม มันใช้สำหรับควบคุมการทำงานในกรณีที่เงื่อนไขเป็นจริง นี่เป็นรูปแบบการใช้งานของคำสั่ง If ในภาษา PHP

```
if (expression)
    statements
```

ตัวอย่างการใช้งาน

```
<?php

$number = 5;

if ($number == 5) {
    echo "Number is equal 5.";
}

?>
```

ในตัวอย่าง เรามีตัวแปร $number ในการเก็บค่าของตัวเลข เนื่องจาก Expression เป็นจริง นั่นคือในตัวแปรมีค่าเท่ากับ 5 โปรแกรมจึงทำงานในบล็อคของคำสั่ง If และแสดงข้อความ "Number is equal 5."

ตัวอย่างเพิ่มเติมสำหรับการใช้คำสั่ง If กับเงื่อนไขที่ซับซ้อนขึ้น


```
<?php

$username = “dech";
$password = "123456";

if ($username == “dech" && $password == "123456") {
    echo "Login succeeded.";
}

?>
```

ในตัวอย่าง เป็นการตรวจสอบการเข้าสู่ระบบของเว็บไซต์ เราได้สร้าง Expression ที่ซับซ้อนขึ้นโดยมีตัวแปร $username และ $password เพื่อให้ในบล็อคคำสั่งทำงานชื่อผู้ใช้ต้องเป็น “dech" และรหัสผ่านต้องเป็น "123456" และเนื่องจากค่าในตัวแปรทำให้ Expression เป็นจริง ทำให้โปรแกรมแสดงข้อความว่า "Login succeeded."

คำสั่ง If Else

- คำสั่ง If Else ใช้สำหรับตรวจสอบเงื่อนไขเช่นเดียวกับคำสั่ง If แต่ในการทำงานจะมีการเพิ่มบล็อคของคำสั่ง else เข้ามาถ้าหากเงื่อนไขในคำสั่ง If ไม่เป็นจริง มาดูตัวอย่างการใช้คำสั่ง If Else ในภาษา PHP

```
<?php

$money = 80;

if ($money >= 100) {
    echo "Buy a video game.";
} else {
    echo "Buy a book.";
}

?>
```

ในตัวอย่างเป็นการใช้งานคำสั่ง If Else กับสถาณการณ์ที่เราได้พูดถึงก่อนหน้านี้ เรามีตัวแปร $money เก็บค่าของจำนวนเงิน ถ้ามีเงินมากกว่าหรือเท่ากับ 100 เหรียญเราจะซื้อวิดีโอเกม แต่ถ้าไม่ใช่จะซื้อหนังสือแทน และเนื่องจากเราได้กำหนดค่าในตัวแปรเพียง 80 โปรแกรมจึงทำงานในบล็อคของคำสั่ง Else แทน

คำสั่ง If Else-If

- ในการใช้กับโปรแกรมคำนวณเกรด

```
<?php
$score = 79;
if ($score >= 80) {
    echo "Your grade is A.";
} else if ($score >= 70) {
    echo "Your grade is B.";
} else if ($score >= 60) {
    echo "Your grade is C.";
} else if ($score >= 50) {
    echo "Your grade is D.";
} else {
    echo "Sorry, you got grade F.";
}
?>
```

ในตัวอย่าง เป็นโปรแกรมคำนวณเกรดโดยการคำนวณจากคะแนนที่มี เราใช้คำสั่งตรวจสอบเงื่อนไข If-Else แบบหลายทางเลือกในการสร้างเงื่อนไขใสแต่ละช่วงคะแนนและเกรดที่จะได้รับ

Your grade is B.
นี่เป็นผลลัพธ์เมื่อรันโปรแกรม เพราะว่าคะแนน 79 อยู่ในช่วงของเกรด B ที่เราได้กำหนดในเงื่อนไข คุณอาจจะลองเปลี่ยนเงื่อนไขเป็นแบบอื่นเพื่อดูผลลัพธ์

คำสั่ง Switch

- ใช้สำหรับเปรียบเทียบกับค่าคงที่โดยตรงที่ไม่ใช่ Expression มาดูตัวอย่างการใช้งาน

```php
<?php
$abb = "th";
switch ($abb) {
    case "de":
        $country = "Germany";
        break;
    case "th":
        $country = "Thailand";
        break;
    case "hu":
        $country = "Hungary";
        break;
    case "tr":
        $country = "Turkey";
        break;
    default:
        $country = "Unknown country";
}
echo "Your country is $country.";
?>
```

ในตัวอย่างเป็นโปรแกรมในการหาชื่อประเทศจากรหัสย่อโดยการใช้คำสั่ง Switch เรามีตัวแปร $abb สำหรับเก็บรหัสย่อของประเทศในโลก ในการใช้งานจะส่งเป็นอากิวเมนต์เข้าไปยังคำสั่ง Switch และโปรแกรมจะทำการตรวจสอบกับเงื่อนไขในแต่ละ case เมื่อเงื่อนไขตรงกับ Case ใดๆ โปรแกรมจะทำงานคำสั่งหลังจาก Case นั้นจนสิ้นสุดบล็อคคำสั่ง Switch เราจำเป็นต้องใช้คำสั่ง break เพื่อหยุดการทำงานของโปรแกรมสำหรับแต่ละ Case

Your country is Thailand.
นี่เป็นผลลัพธ์เมื่อรันโปรแกรม ในตัวแปร $country จะมีค่าเป็น "Thailand" เพราะว่าตรงกับเงื่อนไขใน case "th"

คำสั่ง If ซ้อนกัน

```php
<?php
$name = "Mateo";
$logged_in = true;
$lang = "en";
if ($logged_in) {
    echo "Hello $name, you now logged in.\n";

    if ($lang == "en") {
        echo "The website displayed in English.\n";
    } else if ($lang == "th") {
        echo "The website displayed in Thai.\n";
    } else {
        echo "The language was not set.\n";
    }
} else {
    echo "You are not logged in.\n";
}
?>
```

ในตัวอย่าง เป็นการใช้งานคำสั่งเงื่อนไขซ้อนกัน คำสั่ง If ด้านนอกเป็นการตรวจสอบการเข้าสู่ระบบของเว็บไซต์ ถ้าผู้ใช้เข้าสู่ระบบเราจะแสดงข้อความทักทายจะมีคำสั่ง If ที่ซ้อนกันอยู่ภายในสำหรับตรวจสอบภาษาที่จะแสดงในเว็บไซต์

Hello Mateo, you now logged in.
The website is displayed in English.
นี่เป็นผลลัพธ์เมื่อรันโปรแกรม ซึ่งทำงานในบล็อคของคำสั่งที่ซ้อนกันของ if ($logged_in) และ if ($lang == "en") ตามลำดับ

คำสั่งวนซ้ำ

- ในการเขียนโปรแกรม การวนซ้ำนั้นเป็นสิ่งที่สำคัญในการที่จะทำให้โปรแกรมสามารถทำงานด้วยคำสั่งเดิมได้อย่างมีประสิทธิภาพ ยกตัวอย่าง เช่น การแสดงรายชื่อของสมาชิกของเว็บไซต์จำนวน 20 คน นี่เป็นตัวอย่างที่ต้องใช้คำสั่งวนซ้ำในการจัดการ โดยมีแนวคิดที่ว่าเมื่อคุณสามารถแสดงข้อมูลของสมาชิกคนแรกได้แล้ว คนต่อไปก็สามารถทำในแบบเดียวกันได้

คำสั่ง while loop

- คำสั่ง While loop คือคำสั่งวนซ้ำที่พื้นฐานที่สุดในภาษา PHP มันใช้สำหรับควบคุมการทำงานของโปรแกรมให้ทำงานซ้ำๆ ภายใต้เงื่อนไขที่กำหนด นี่เป็นรูปแบบของการใช้งานคำสั่ง While loop ในภาษา PHP

- while (expression)
- statements

คำสั่ง while loop

- ในการทำงานของคำสั่ง While loop จะทำงานในขณะที่ expression เป็นจริง ซึ่ง statements เป็นคำสั่งภายในบล็อคของ While loop ที่อาจจะประกอบไปด้วยหนึ่งหรือหลายคำสั่ง มาดูตัวอย่างการใช้งานคำสั่ง While loop ในภาษา PHP

```
<?php

$i = 1;

while ($i <= 10) {
    echo "$i\n";
    $i++;
}

?>
```

ในตัวอย่างเป็นการใช้คำสั่ง While loop ในการแสดงผลตัวเลขจาก 1 - 10 ตัวแปร $i เป็นตัวแปรกำหนดค่าเริ่มต้นในการทำงาน และ $i <= 10 เป็นเงื่อนไขของคำสั่งวนซ้ำ ซึ่งหมายความว่าโปรแกรมจะทำงานในบล็อคของคำสั่งถ้าหากค่าของ $i น้อยกว่าหรือเท่ากับ 10 ในบล็อคของคำสั่ง While loop เราแสดงผลตัวเลขและเพิ่มค่า $i ในแต่ละรอบ

คำสั่ง do while loop

- คำสั่ง Do while loop นั้นมีการทำงานคล้ายกับคำสั่ง While loop แต่สิ่งที่แตกต่างคือในคำสั่ง Do while loop จะทำงานภายใน Loop ก่อนอย่างน้อยหนึ่งรอบและตรวจสอบเงื่อนไขในภายหลัง นี่เป็นรูปแบบของการใช้งานคำสั่ง Do while loop ในภาษา PHP

```
do {
    statements
} while (expression);
```

คำสั่ง do while loop

- ในการใช้งานคำสั่ง Do while loop นั้นส่วนของการตรวจสอบเงื่อนไข expression จะอยู่ตอนท้าย นั่นหมายความจะต้องมีการทำงานในลูปอย่างน้อยแน่นอน 1 รอบ มาดูตัวอย่างการใช้งานคำสั่ง Do while loop ในภาษา PHP กับโปรแกรมการสุ่มตัวเลข

```php
<?php

do {
    $number = rand(0, 8);
    echo "Random number $number\n";
} while ($number != 0);

?>
```

ในตัวอย่าง เป็นการใช้คำสั่ง Do while loop ในการสุ่มตัวเลขระหว่าง 0 - 8 และโปรแกรมจะทำงานใน Loop ในขณะที่ค่าที่สุ่มได้ไม่ใช่ 0 ในเงื่อนไข $number != 0

อาเรย์ (Array)

- อาเรย์ (Array) คือประเภทข้อมูลที่เก็บข้อมูลเป็นชุดลำดับเรียงต่อกันในหน่วยความจำ อาเรย์เป็นตัวแปรประเภทหนึ่งในภาษา PHP ที่สามารถเก็บข้อมูลได้มากกว่าหนึ่งค่า อาเรย์ช่วยอำนวยความสะดวกในกรณีที่เราต้องการจัดการข้อมูลประเภทเดียวกันเป็นจำนวนมาก ยกตัวอย่างเช่น คุณต้องการเก็บคะแนนของนักเรียน 10 คน การใช้อาเรย์จึงเป็นสิ่งที่สะดวก

ประกาศและใช้งานอาเรย์

- ก่อนเริ่มใช้งานอาเรย์ เราจะให้คุณเห็นถึงความแตกต่างระหว่างการใช้งานอาเรย์และไม่ใช้ สำหรับจัดการข้อมูลจำนวนมาก คุณมีตัวเลข 5 ตัวที่ต้องการเก็บในตัวแปรและนำตัวเลขเหล่านั้นมาคำนวณ

```
$number1 = 10;
$number2 = 20;
$number3 = 30;
$number4 = 40;
$number5 = 50;
```

- คำตอบก็คือคุณต้องประกาศตัวแปร 5 ตัวสำหรับเก็บข้อมูลเหล่านี้ ในตอนนี้เราจะใช้อาเรย์สำหรับเก็บข้อมูลเหล่านี้แทน

```
$numbers = [10, 20, 30, 40, 50];
```

ตัวอย่างการประกาศอาเรย์ในรูปแบบต่างๆ ที่ทำได้ในภาษา PHP

- $names = [“เดช", “ดาว", “เปา", “พี่เอ๋"];
- $mixed = [1, "PHP", "C#", 1.54, true];
- $empty = [];

การใช้คำสั่ง For loop กับอาเรย์

- เนื่องจากอาเรย์นั้นเก็บข้อมูลเป็นลำดับและมีการเข้าถึงผ่าน Index ดังนั้นคำสั่ง For loop กับอาเรย์จึงเป็นสิ่งที่จำเป็นในการเขียนโปรแกรม ซึ่งโดยทั่วไปแล้วในการเขียนโปรแกรม เรามักจะใช้คำสั่ง For loop กับอาเรย์เสมอ

```php
<?php
$numbers = [];

for ($i = 0; $i < 10; $i++) {
    $numbers[] = rand(1, 100);
}
echo "Random numbers: ";
for ($i = 0; $i < 10; $i++) {
    echo $numbers[$i], ", ";
}
?>
```

ในตัวอย่างเป็นการใช้คำสั่ง For loop กับอาเรย์ ใน Loop แรก เราได้ทำการวนรอบการสุ่มตัวเลขระหว่าง 1 - 100 ด้วยฟังก์ชัน rand() จำนวน 10 ตัวและใส่ในอาเรย์ $numbers และใน For loop ที่สองเป็นการวนอ่านค่าภายในอาเรย์ เราใช้ตัวแปร $i เป็น Index ในการเข้าถึงค่าภายในอาเรย์ $numbers[$i]

Random numbers: 7, 92, 39, 32, 75, 67, 23, 29, 8, 73, นี่เป็นผลลัพธ์เมื่อรันโปรแกรม ตัวเลขนั้นได้จากการสุ่มซึ่งอาจจะแตกต่างกันในการรันแต่ละครั้ง

อาเรย์ 2 มิติ

- ที่ผ่านมาเป็นการใช้งานอาเรย์หนึ่งมิติ ในภาษา PHP คุณสามารถสร้างอาเรย์หลายมิติได้ หรือเราเรียกว่าอาเรย์ของอาเรย์ ลองจินตนาการว่าคุณใส่อาเรย์ลงไปในอาเรย์ ซึ่งต่อไปนี้เราจะพูดเกี่ยวกับการประกาศและใช้งานอาเรย์ใน 2 มิติ

- $two_dimension_array = [Array, Array, Array, ...];

- อาเรย์ 2 มิตินั้นอาจจะใช้กับงานที่หลากหลายและแตกต่างกันออกไป รูปแบบการเก็บข้อมูลของมันเหมือนตารางหรือเมทริกซ์ ซึ่งประกอบไปด้วยแถวและคอลัมน์

ตัวอย่างการประกาศอาเรย์ 2 มิติในภาษา PHP

```
<?php
$number = [
    [43, 23, 5, 43],
    [13, 21, 63, 93],
    [54, 65, 82, 27],
    [8, 36, 73, 82],
];
print_r($number);
?>
```

- ในตัวอย่าง เป็นการประกาศอาเรย์ 2 มิติ การในเข้าถึงข้อมูลของอาเรย์ เราเข้าถึงโดยใช้ Index สองตัว เช่น ต้องการเข้าถึงตัวเลข 21 ในแถวที่สอง จะเข้าถึงโดย $number[1][1] และเข้าถึงตัวเลข 73 ในแถวสุดท้ายจะเป็น <code>$number[3][2] เป็นต้น

อาเรย์ 3 มิติและหลายมิติ

- นอกจากนี้คุณยังสามารถสร้างอาเรย์ที่มากกว่า 2 มิติได้ เช่น การสร้างอาเรย์ 3 มิติ คุณก็ใส่อาเรย์ 2 มิติลงไปในอาเรย์ 3 มิติเป็นต้น แต่โดยส่วนมากแล้วในการเขียนโปรแกรม เราใช้มากสุดเพียงแค่ 3 มิติ

```
<?php

$two_dimension_array = [Array, Array, ...];
$three_dimension_array = [$two_dimension_array,
$two_dimension_array, ... ];
...

?>
```

ฟังก์ชัน

- ฟังก์ชัน (Functions) คือส่วนของโปรแกรมหรือซอสโค้ดที่ใช้สำหรับจัดการกับงานที่เฉพาะเจาะจง ฟังก์ชันเป็นขั้นตอนของการทำงานบางอย่างเพื่อให้ได้ผลลัพธ์ตามที่ต้องการโดยการเรียกใช้ฟังก์ชัน ในภาษา PHP มีฟังก์ชันมาตรฐานที่คุณสามารถใช้งานได้ ซึ่งเรียกว่า Predefined function ซึ่งแน่นอนฟังก์ชันเหล่านี้มีไว้สำหรับจัดการงานทั่วไปเท่านั้น

- ฟังก์ชันมีส่วนประกอบสองอย่างคือส่วนหัวของฟังก์ชัน (head) และส่วนการทำงานของฟังก์ชัน (function body) ในภาษา PHP เรากำหนดจำนวนของพารามิเตอร์ของฟังก์ชันที่ส่วนหัว และการทำงานของฟังก์ชันในส่วนการทำงาน ซึ่งฟังก์ชันอาจจะมีการส่งค่ากลับหรือไม่มีก็ได้ (return) นี่เป็นรูปแบบของการประกาศฟังก์ชันในภาษา PHP

```
function name(parameters) {
    statements
    return value (optional)
}
```

ในการสร้างฟังก์ชันในภาษา PHP เราใช้คำสั่ง function และตามด้วยชื่อของฟังก์ชันซึ่งมีหลักการตั้งชื่อเหมือนกับตัวแปร parameters เป็นการกำหนดพารามิเตอร์ของฟังก์ชัน ซึ่งอาจจะมีหรือไม่มีก็ได้ ภายในบล็อคการทำงานของฟังก์ชัน {} เป็นขั้นตอนการทำงานของฟังก์ชัน และคำสั่ง return ใช้สำหรับส่งค่ากลับซึ่งเป็นทางเลือก

การสร้างฟังก์ชันในภาษา PHP

- การสร้างฟังก์ชันเป็นการรวบรวมกลุ่มคำสั่งของโปรแกรมในการทำงานที่เฉพาะเจาะจง การสร้างฟังก์ชันจะทำให้คุณสามารถเรียกใช้โค้ดเดิม (reuse) โดยที่ไม่ต้องเขียนขึ้นใหม่ มาดูตัวอย่างการสร้างฟังก์ชันในภาษา PHP

```
function welcome() {
    echo "Welcome to Mr.Dech Homepage .\n";
}

function hello($name) {
    echo "Hello $name!\n";
}

function add_number($a, $b) {
    $sum = $a + $b;
    echo "$a + $b = $sum\n";
}
```

ในตัวอย่างเป็นการสร้างฟังก์ชันในภาษา PHP ซึ่งเราได้สร้าง 3 ฟังก์ชันสำหรับทำงานที่แตกต่างกันออกไป ฟังก์ชัน welcome() เป็นฟังก์ชันสำหรับแสดงข้อความต้อนรับสำหรับ Mr.Dech Homepage

ฟังก์ชัน hello() เป็นฟังก์ชันในการกล่าวคำทักทาย โดยมี $name ที่ส่งเข้ามายังฟังก์ชัน เรียกว่าพารามิเตอร์ของฟังก์ชัน และฟังก์ชัน add_number() เป็นฟังก์ชันสำหรับบวกตัวเลขสองตัวที่ส่งเป็นพารามิเตอร์คือ $a และ $b และแสดงผลรวมออกทางหน้าจอ ทั้ง 3 ฟังก์ชันไม่มีการส่งค่ากลับ

ฟังก์ชันที่มีการส่งค่ากลับ

```
function square_area($width, $height) {
    return $width * $height;
}
```

- ในตัวอย่างเป็นฟังก์ชันที่มีการส่งค่ากลับในรูปแบบต่างๆ การส่งค่ากลับของฟังก์ชันนั้นจะใช้คำสั่ง return และตามด้วยค่าที่ต้องการส่งกลับ ในฟังก์ชันอาจจะมีการใช้คำสั่ง return ได้หลายที่ ซึ่งเมื่อโปรแกรมพบกับคำสั่งนี้ จะเป็นการจบการทำงานในฟังก์ชันในทันที

การเรียกใช้งานฟังก์ชัน

```
<?php
function hello($name) {
    echo "Hello $name.\n";
}
function add_number($a, $b) {
    $sum = $a + $b;
    echo "$a + $b = $sum\n";
}
welcome();
hello(“เดช");
hello(“ดาว");
hello(“Ake");
add_number(2, 3);
add_number(34, 15);
add_number(1.45, 2.4);
?>
```

ตัวอย่างการใช้งานฟังก์ชันที่มีการส่งค่ากลับในภาษา PHP


```php
<?php

function get_pronoun($gender) {
    if ($gender == 0) {
        return "he";
    } else {
        return "she";
    }
}

function celsius_to_fahrenheit($celsius) {
    return $celsius * 1.8 + 32;
}

function mph_to_kmph($mph) {
    return $mph * 1.60934;
}
```

```php
echo "It was " . celsius_to_fahrenheit(3) . " degree yesterday, ";
echo "the weather was " . get_weather(3) . ".\n";

echo "It is " . celsius_to_fahrenheit(12) . " degree today, ";
echo "the weather is " . get_weather(12) . ".\n";

echo "It will be " . celsius_to_fahrenheit(22) . " degree
tomorrow, ";
echo "the weather would get " . get_weather(22) . "er.\n";

$km = mph_to_kmph(36);
echo "The car is running at 36 mph or $km km/h.\n";

?>
```

ออบเจ็ค ในภาษา PHP

- คือตัวแปรประเภทหนึ่งที่สร้างมาจากคลาสหรือ Class Instance ออบเจ็คมีการอ้างถึงสมาชิกของมันที่เป็นตัวแปรและเมธอด ซึ่งในภาษา PHP นั้นทุกอย่างต่างก็เป็นออบเจ็คไม่ว่าจะเป็น ตัวแปร อาเรย์ หรือฟังก์ชัน และคลาสเป็นเหมือนประเภทข้อมูลที่นำมาสร้างออบเจ็ค ดังนั้นออบเจ็คหลายออบเจ็คอาจจะมีการสร้างมาจากคลาสเดียวกัน (Class base)

การสร้างออบเจ็ค

```php
<?php
class User {
    public $firstName;
    public $lastName;
    public $homeTown;
    private $birthYear;

    public function indroduce() {
        echo "I'm $this->firstName $this->lastName\n";
        echo "I live in $this->homeTown\n";
    }
    public function setBirthYear($birthYear) {
        $this->birthYear = $birthYear;
    }
    public function getAge() {
        $currentYear = 2017;
        $age = $currentYear - $this->birthYear;
        return $age;
    }
}
```

```php
$user1 = new User();
$user1->firstName = “Dech";
$user1->lastName = “Thammasiri";
$user1->homeTown = “Pathumtanee";
$user1->setBirthYear(1977);

$user1->indroduce();
echo "My age is " . $user1->getAge() . "\n";

$user2 = new User();
$user2->firstName = “Devid";
$user2->lastName = “Backham";
$user2->homeTown = “Manchester United";
$user2->setBirthYear(1980);

echo "Name: $user2->firstName $user2->lastName\n";
echo "Home town: $user2->homeTown\n";
echo "Age: " . $user2->getAge() . "\n";

?>
```

การสร้างและจัดการ Form

- Element <form> ใช้ในการสร้างแบบฟอร์มกรอกข้อมูล

<form>

.form elements

.</form>

ภายใน Element <form> จะประกอบด้วย Element ต่างๆ ในการสร้างช่องกรอกข้อมูล

ซึ่ง Element ประเภทต่างๆ ประกอบด้วย ช่องกรอกข้อความ กล่องติ๊กถูก ตัวเลือก ปุ่ม Submit และอื่นๆ

Element: <input>

- <input> เป็น Element ที่สำคัญที่สุด
- <input> สามารถใช้สร้างช่องกรอกข้อมูลได้หลายแบบ ขึ้นอยู่กับประเภทที่เรากำหนดใน Attribute type
- ตัวอย่าง:

ประเภท	อธิบาย
<input type="text">	สร้างกล่องกรอกข้อความบรรทัดเดียว (Text Input)
<input type="radio">	สร้างตัวเลือกแบบเลือกข้อใดข้อหนึ่ง (radio button)
<input type="submit">	สร้างปุ่ม Submit
<input type=“reset">	สร้างปุ่ม reset

Attribute: action

Attribute action ใช้ในการระบุว่าเมื่อ submit แบบฟอร์มแล้วจะส่งข้อมูลไปที่ไหน ซึ่งปกติก็เป็นหน้าเว็บอีกหน้าหนึ่งที่มีโค้ดในการรับไฟล์ไปประมวลผล

อย่างในตัวอย่าง แบบฟอร์มจะส่งข้อมูลไปยังหน้าเว็บที่ชื่อว่า "action_page.php" ซึ่งเป็นโค้ดฝั่งเซิฟเวอร์รับข้อมูลไปทำงานต่อ

```
<form action="action_page.php“ >
...
</form>
```

Attribute: method

Attribute method ใช้ในการระบุวิธีการส่งค่าของ http ว่าจะให้ส่งแบบ GET หรือแบบ POST เมื่อ submit ฟอร์มแล้ว

<form action="action_page.php" method="GET">

หรือ

<form action="action_page.php" method="POST">

เมื่อไหร่ใช้ GET ?

ค่าเริ่มต้นของการส่งข้อมูลคือแบบ GET อยู่แล้ว

การส่งข้อมูลด้วยวิธี GET คือการส่งข้อมูลผ่าน URL เลย เมื่อ submit แบบฟอร์ม ข้อมูลที่กรอกในฟอร์มจะมองเห็นได้บน URL ของหน้าเว็บ

action_page.php?firstname=Dech&lastname=Thammasiri

การรับค่าในหน้าต่อไป

```
$firstname=$_GET['firstname'];
$lastname=$_GET['lastname'];
```

เมื่อไหร่ใช้ POST ?

- คุณควรใช้ POST กับการกรอกข้อมูลที่มีความสำคัญหรือข้อมูลส่วนตัว การส่งแบบ POST เมื่อกด Submit จะไม่แสดงข้อมูลใดๆบน URL ของหน้าเว็บ และไม่จำกัดความยาวของข้อมูลอีกด้วย

การรับค่าในหน้าต่อไป

```
$firstname=$_POST['firstname'];
$lastname=$_POST['lastname'];
```

ทดลองปฏิบัติกัน

มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

ขอบคุณครับ


มหาวิทยาลัยราชภัฏนครปฐม
NAKHON PATHOM RAJABHAT UNIVERSITY