บทที่ 3

ภาษา PHP

PHP ย่อมาจาก **P**HP **H**ypertext **P**reprocessor แต่เดิมย่อมาจาก Personal Home Page Tools PHP คือภาษาคอมพิวเตอร์จำพวก scripting language ภาษาจำพวกนี้คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า Script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ตัวอย่างของภาษาสคริปก็เช่น JavaScript , Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า Server-side หรือ HTML-embedded scripting language นั้นคือในทุกๆ ครั้งก่อนที่เครื่องคอมพิวเตอร์ซึ่งให้บริการเป็น Web server จะส่งหน้าเว็บเพจที่เขียนด้วย PHP ให้เรา มันจะทำการประมวลผลตามคำสั่งที่มีอยู่ให้เสร็จเสียก่อน แล้วจึงค่อยส่งผลลัพธ์ที่ได้ให้เรา ผลลัพธ์ที่ได้นั้นก็คือเว็บเพจที่เราเห็นนั่นเอง ถือได้ว่า PHP เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้เราสามารถสร้าง Dynamic Web pages (เว็บเพจที่มีการโต้ตอบกับผู้ใช้) ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

ประวัติความเป็นมาของภาษา PHP

PHP เกิดในปี 1994 โดย Rasmus Lerdorf โปรแกรมเมอร์อเมริกันได้คิดค้นสร้างเครื่องมือที่ใช้ในการพัฒนาเว็บส่วนตัวของเขา โดยใช้ข้อดีของภาษา C และ Perl เรียกว่า Personal Home Page และได้สร้างส่วนติดต่อกับฐานข้อมูลที่ชื่อว่า Form Interpreter (FI) รวมทั้งสองส่วน เรียกว่า PHP/FI ซึ่งก็เป็นจุดเริ่มต้นของ PHP มีคนที่เข้ามาเยี่ยมชมเว็บไซต์ของเขาแล้วเกิดชอบจึงติดต่อขอเอาโค้ดไปใช้บ้างและนำไปพัฒนาต่อ ในลักษณะของ Open Source ภายหลังมีความนิยมขึ้นเป็นอย่างมากภายใน 3 ปีมีเว็บไซต์ที่ใช้ PHP/FI ในติดต่อฐานข้อมูลและแสดงผลแบบ ไดนามิกและอื่นๆ มากกว่า 50,000 ไซต์

PHP2 (ในตอนนั้นใช้ชื่อว่า PHP/FI) ในช่วงระหว่าง 1995-1997 Rasmus Lerdorf ได้มีผู้ที่มาช่วยพัฒนาอีก 2 คนคือ Zeev Suraski และAndi Gutmans ชาวอิสราเอล ซึ่งปรับปรุงโค้ดของ Lerdorf ใหม่โดยใช้ C++ ให้มีความสามารถจัดการเกี่ยวกับแบบฟอร์มข้อมูลที่ถูกสร้างมาจากภาษา HTML และสนับสนุนการติดต่อกับโปรแกรมจัดการฐานข้อมูล mySQL จึงทำให้ PHP เริ่มถูกใช้มากขึ้นอย่างรวดเร็ว และเริ่มมีผู้สนับสนุนการใช้งาน PHP มากขึ้น โดยในปลายปี 1996 PHP ถูกนำไปใช้ประมาณ 15,000 เว็บทั่วโลก และเพิ่มจำนวนขึ้นเรื่อยๆ ต่อมาก็มผู้เข้ามาช่วยพัฒนาอีก 3 คน คือ Stig Bakken รับผิดชอบความสามารถในการติดต่อ Oracle, Shane Caraveo รับผิดชอบดูแลPHP บน Window 9x/NT, และ Jim Winstead รับผิดชอบการตรวจความบกพร่องต่างๆ และได้เปลี่ยนชื่อเป็น Professional Home Pageในเวอร์ชั่นที่ 2

PHP3 ออกมาในช่วงระหว่างเดือน มิถุนายน 1997 ถึง 1999 ได้ออกสู่สายตาของนักโปรแกรมเมอร์มีคุณสมบัติเด่นคือสนับสนุนระบบปฏิบัติการทั้ง Window 95/98/ME/NT, Linux และเว็บเซร์ฟเวอร์ อย่าง IIS, PWS, Apache, OmniHTTPd สนับสนุน ระบบฐานข้อมูลได้หลายรูปแบบเช่น SQL Server, MySQL, mSQL, Oracle, Informix, ODBC

PHP4 ตั้งแต่ 1999 - 2007 ซึ่งได้เพิ่ม Functions การทำงานในด้านต่างๆให้มากและง่ายขึ้นโดยบริษัท Zend ซึ่งมี Zeev และ Andi Gutmans ได้ร่วมก่อตั้งขึ้น (http://www.zend.com) ในเวอร์ชั่นนี้จะเป็น compile script ซึ่งในเวอร์ชั่นหน้านี้จะเป็น embed script interpreter ในปัจจุบันมีคนได้ใช้ PHP สูงกว่า 5,100,000 ไซต์ แล้วทั่วโลก และ ผู้พัฒนาได้ตั้งชื่อของ PHP ใหม่ว่า PHP: Hypertext Preprocessor ซึ่งหมายถึงมีประสิทธิภาพระดับโปรเฟสเซอร์สำหรับไฮเปอร์เท็กซ์

PHP5 ตั้งแต่ 2007-ปัจจุบัน มี ได้เพิ่ม Functions การทำงานในด้านต่าง ๆ เช่น

- Object Oriented Model
- การกำหนดสโคป public/private/protected
- Exception handling
- XML และ Web Service
- MySQLi และ SQLite
- Zend Engine 2.0

https://arit.rmutsv.ac.th/th/blogs/52-ประวัติความเป็นมาของ-php-152

ไฟล์ PHP

ไฟล์ PHP สามารถประกอบด้วยข้อความ HTML, CSS, JavaScript และโค้ด PHP

โค้ด PHP จะทำงานบนเซิร์ฟเวอร์และผลลัพธ์จะถูกส่งกลับไปยังเบราว์เซอร์เป็น HTML แบบธรรมดา

ไฟล์ PHP มีนามสกุล ".php"

PHP สามารถทำหน้าที่หลายอย่าง เช่น

สามารถสร้างเนื้อหาแบบไดนามิก

สามารถสร้างเปิดอ่านเขียนลบและปิดไฟล์บนเซิร์ฟเวอร์ได้

สามารถเก็บข้อมูลฟอร์มได้

สามารถส่งและรับคุกกี้ได้

สามารถเพิ่มลบแก้ไขข้อมูลในฐานข้อมูลของคุณได้

สามารถใช้เพื่อควบคุมการเข้าถึงของผู้ใช้

สามารถเข้ารหัสข้อมูล

ด้วย PHP ไม่จำกัดเฉพาะ HTML ที่ส่งออก สามารถส่งออกภาพไฟล์ PDF และแม้แต่ภาพยนตร์ Flash สามารถส่งออกข้อความใดๆ ได้ เช่น XHTML และ XML เป็นต้น

ทำไมต้องเป็น PHP?

PHP ทำงานบนแพลตฟอร์มต่างๆ (Windows, Linux, Unix, Mac OS X เป็นต้น)

PHP เข้ากันได้กับเซิร์ฟเวอร์เกือบทั้งหมดที่ใช้ในปัจจุบัน (Apache, IIS, ฯลฯ)

PHP สนับสนุนฐานข้อมูลหลากหลาย

PHP ฟรี ดาวน์โหลดได้จากแหล่งข้อมูล PHP ได้จาก: www.php.net

PHP เป็นเรื่องง่ายที่จะเรียนรู้และทำงานได้อย่างมีประสิทธิภาพด้านเซิร์ฟเวอร์

การติดตั้ง PHP5

ก่อนการใช้งาน PHP สามารถตรวจสอบเครื่องคอมพิวเตอร์ดังนี้

1. ค้นหาเว็บโฮสต์ที่มีการสนับสนุน PHP และ MySQL
2. ติดตั้งเว็บเซิร์ฟเวอร์บนพีซีจากนั้นติดตั้ง PHP และ MySQL

.....เพิ่มเติม

รูปแบบของ PHP

- PHP จะทำงานบนเซิร์ฟเวอร์ (Server) และถูกส่งกลับไปยังเบราเซอร์ (Browser) ในรูปแบบโค๊ด HTML (Hypertext Markup Language)

- สคริปต์ PHP สามารถวางไว้ที่ใดก็ได้ในไฟล์ และเริ่มต้นด้วย <?php และลงท้ายด้วย ?> แสดงดังนี้

```
<?php
    //code
?>
```

- เครื่องหมายอัฒภาค (;) คือ เครื่องหมายที่ใช้สิ้นสุดคำสั่งในแต่ละบรรทัดแสดงดังตัวอย่าง

```
<?php
    echo "hello";
?>
```

- คำอธิบาย (Comment) คือ การอธิบายคำสั่งในแต่ละบรรทัด เพื่ออธิบายโค๊ด (Code) ในส่วนที่ต้องการอธิบาย และโปรแกรมจะไม่นำมาประมวลผลหรือบรรทัดที่มีคอมเม้นจะไม่ทำงานแสดงดังตัวอย่างดังนี้

```
<?PHP
    // เป็นคอมเม้นที่ทำงานเพียงบรรทัดเดียว
    # เป็นคอมเม้นที่ทำงานเพียงบรรทัดเดียว
    /* เป็นคอมเม้นที่ทำงานหลายบรรทัด */
    echo "hello";
?>
```

ตัวแปร (Variable)

ตัวแปร คือ สิ่งที่ใช้เก็บค่าของข้อมูลในหน่วยความจำ เพื่อนำไปประมวลผล ซึ่งตัวแปรจะขึ้นต้นด้วยเครื่องหมายดอลลา $ และตามด้วยชื่อของตัวแปร ตัวอย่างเช่น `$x = 5;` และมีกฎการตั้งชื่อตัวแปรดังต่อไปนี้

- ขึ้นต้นด้วย $
- ตามด้วย A-Z หรือ a-z หรือ 0-9 หรือ _ เช่น $chalawan; $Chalawan_007;
- ตัวแปรที่ขึ้นต้นด้วยตัวพิมพ์ใหญ่ หรือตัวพิมพ์เล็กถือเป็นคนละตัว เช่น $Dog; $dog;
- ไม่ตั้งชื่อซ้ำคำสงวน เช่น $_POST; $_SESSION; $_GET;

- คำสงวน (Reserved Words) หมายถึง คำที่จะใช้เป็นคำสั่งเฉพาะของ PHP ดังนั้นเราต้องไม่นำคำเหล่านี้มาตั้งเป็นชื่อตัวแปร หรือฟังก์ชัน มิฉะนั้นจะทำให้เกิดข้อผิดพลาดขึ้นได้ แสดงดังภาพที่ 1

and	false	or	extends	__FILE__
break	for	require	not	__METHOD__
case	foreach	return	vitual	use
class	function	static	xor	clone(5)
continue	global	switch	while	old_fuctinon(4 only)
do	if	this	date	thow(5)
else	include	true	time	protected(5)
elseif	new	var	explode	catch(5)
try(5)	public	nal	var	new

ภาพที่ 1 คำสงวนในภาษา PHP

ที่มา : https://www.itgenius.co.th/article/คำสงวนของ%20PHP%20(PHP%20Reserved%20Words).html

คำสั่งแสดงผล

เมื่อต้องการแสดงผลบนหน้าจอ ในภาษา PHP มี 2 คำสั่ง ได้แก่ echo และ print ความแตกต่างของทั้ง 2 ฟังก์ชั่นสามารถกล่าวได้ดังนี้

1. return value คือ การคืนค่าของคำสั่ง echo จะไม่มีค่า return ผลของคำสั่ง ส่วนคำสั่ง print จะมีการคืนค่ามาเป็น true, false
2. ความเร็วในการทำงานของ echo จะสามารถทำงานได้เร็วกว่า print เพราะ echo ไม่มีการคืนค่าของคำสั่ง
3. คำสั่ง print ทำงานเหมือนฟังก์ชั่น ทำให้ print ใช้ในรูปประโยคที่ซับซ้อนได้ ส่วน echo ไม่สามารถทำได้ เช่น

```php
<?php
    $age?print "true":print "false";
?>
```

ตัวอย่างการใช้ echo

```php
<?php
    ECHO "Hello World!<br>";
    echo "Hello World!<br>";
    EcHo "Hello World!<br>";
?>
```

ตัวอย่างการใช้ print

```php
<?php
    print "Hello World!<br>";
?>
```

ชนิดของข้อมูล (Data Type)

ตัวแปรสามารถจัดเก็บข้อมูลประเภทต่างๆ และชนิดข้อมูลที่แตกต่างกันสามารถทำสิ่งต่างๆ แสดงตารางดังต่อไปนี้

ประเภทตัวแปร	ความหมาย
String	ตัวอักษรที่นำไปคำนวณทางคณิตศาสตร์ไม่ได้
Integer	เลขจำนวนเต็ม
Float	เลขจำนวนจริงเป็นทศนิยม
Boolean	ข้อมูลที่เก็บค่าความเป็นจริง คือ TRUE กับค่าความเป็นเท็จ คือ FALSE
Array	ตัวแปรชุด
Object	เก็บคุณสมบัติของวัตถุ (Object)
Resource	สำหรับอ้างอิงถึงแหล่งภายนอก เช่น การเปิดไฟล์ข้อมูล การเชื่อมต่อฐานข้อมูล
Null	ตัวแปรที่ไม่มีค่าอะไรเลยเรียกว่ามีค่าเป็น Null เช่น เมื่อประกาศตัวแปรแล้วแต่ยังไม่ได้กำหนดค่าใดๆให้ตัวแปร กำหนดค่าให้ตัวแปรมีค่าเป็น Null $MySalary =

	NULL;

String คือ ตัวอักษรที่นำไปคำนวณทางคณิตศาสตร์ไม่ได้ เป็นข้อความใดก็ได้ภายใต้เครื่องหมายคำพูด (“ ” หรือ ‘ ’) **ดังตัวอย่าง**

```php
<?php
    $x = "Hello world!";
    $y = 'Hello world!';
    echo $x;
    echo "<br>";
    echo $y;
?>
```

Integer คือ เลขจำนวนเต็มหรือ ชนิดข้อมูลจำนวนเต็มเป็นตัวเลขที่ไม่ใช่ทศนิยมระหว่าง -2,14,483,648 และ 2,147,483,647 มีกฎเกณฑ์ดังนี้

- จำนวนเต็มต้องมีอย่างน้อยหนึ่งหลัก
- จำนวนเต็มต้องไม่มีจุดทศนิยม
- จำนวนเต็มสามารถเป็นได้ทั้งบวกหรือลบ
- จำนวนเต็มสามารถระบุได้ในรูปแบบสามรูปแบบคือทศนิยม (ทศนิยม 10 ตัว) เลขฐานสิบหก (อิงเลขฐานสิบหก - นำหน้าด้วย 0x) หรือเลขฐานแปด (ตามที่ 8 - นำหน้าด้วย 0)

ตัวอย่างดังนี้

```php
<?php
    $x = 254;
    var_dump($x);
?>
```

Float คือ เลขจำนวนจริงเป็นทศนิยม **เช่น** `$x = 53.78;`

Boolean คือ ข้อมูลที่เก็บค่าความเป็นจริง คือ TRUE กับค่าความเป็นเท็จ คือ FALSE เช่น

```php
$a = TRUE; $b = false;
```

Array คือ ตัวแปรชุดเก็บค่าของข้อมูลได้หลายค่า เช่น

```php
$cars = array("Volvo","BMW","Toyota");
```

Object คือ ตัวแปรประเภทหนึ่งที่สร้างมาจากคลาสหรือ *Class Instance* ออบเจ็คมีการอ้างถึงสมาชิกของมันที่เป็นตัวแปรและเมธอด

ตัวอย่าง

```php
<?php
    class Car {
        function Car() {
            $this->model = "VW";
        }
    }

    // create an object
    $herbie = new Car();

    // show object properties
    echo $herbie->model;
?>
```

Resource คือ สำหรับอ้างอิงถึงแหล่งภายนอก เช่น การเปิดไฟล์ข้อมูล การเชื่อมต่อฐานข้อมูล

Null คือ ตัวแปรที่ไม่มีค่าอะไรเลยเรียกว่ามีค่าเป็น Null เช่น เมื่อประกาศตัวแปรแล้วแต่ยังไม่ได้กำหนดค่าใดๆให้ตัวแปร กำหนดค่าให้ตัวแปรมีค่าเป็น Null $MySalary = NULL;

ตัวอย่างการประกาศตัวแปร

โค๊ด	ผลลัพท์
`<?php` `  $name = "Chalawan Wantong";` `  $age = 5;` `  $GPA = 2.43;` `  echo "$name  ";` `  echo "$age  ";` `  echo "$GPA ";` `?>`	Chalawan Wantong 5 2.43

ตัวดำเนินการ (Operators)

ใช้เพื่อดำเนินการกับตัวแปรและค่าต่างๆ ในภาษา PHP จะทำการแบ่งกลุ่มของ Operators ไว้ดังนี้

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)
- ตัวดำเนินการกำหนดค่า (Assignment operators)
- ตัวดำเนินการเชิงเปรียบเทียบ (Comparison operators)
- ตัวดำเนินการเพิ่มค่าหรือลดค่า (Increment/Decrement operators)
- ตัวดำเนินการเชิงตรรกะ (Logical operators)
- ตัวดำเนินการเชิงข้อความ (String operators)
- ตัวดำเนินการเชิงอาร์เรย์ (Array operators)

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)

ตัวดำเนินการเลขคณิตศาสตร์ ใช้กับค่าตัวเลขเพื่อดำเนินการทางคณิตศาสตร์ทั่วไป เช่นการบวก การลบ การคูณ เป็นต้น

ตัวดำเนินการ	ชื่อ	**ตัวอย่าง** กำหนดให้ `$x = 5; $y =2;`	ผลลัพธ์
+	บวก	`echo $x + $y;`	7
-	ลบ	`echo $x - $y;`	3
*	คูณ	`echo $x * $y;`	10
/	หาร	`echo $x / $y;`	2.5
%	หารเอาค่าเศษ	`echo $x % $y;`	1

ตัวดำเนินการกำหนดค่า (Assignment operators)

ตัวดำเนินการกำหนดค่า เป็น operators ที่ใช้สำหรับกำหนดค่าให้กับตัวแปร เช่น $x = $y หมายความว่า กำหนดให้ค่าตัวแปร $x มีค่าเท่ากับค่าของ $y

ตัวดำเนินการ	ความหมาย	ตัวอย่าง กำหนดให้ `$x = 5; $y =2;`	ผลลัพธ์
x = y	x = y	`echo $x = $y;`	2
x += y	x = x + y	`echo $x += $y;`	7
x -= y	x = x - y	`echo $x -= $y;`	3
x *= y	x = x * y	`echo $x *= $y;`	10
x /= y	x = x / y	`echo $x /= $y;`	2.5
x % = y	x = x % y	`echo $x %= $y;`	1

ตัวดำเนินการเชิงเปรียบเทียบ (Comparison operators)

ตัวดำเนินการเชิงเปรียบเทียบ ใช้เพื่อเปรียบเทียบค่าสองค่า และแสดงผลเป็น จริง (true) และเท็จ (false) เท่านั้น

ตัวดำเนินการ	ความหมาย	ตัวอย่าง กำหนดให้
		`$x = 5; $y =2;`
		`$z = '5';`
			ผลลัพธ์

==	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าเท่ากัน	`echo $x == $z;`	true
===	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าเท่ากัน **และ** ประเภทตัวแปรเหมือนกัน	`echo $x === $z;`	false
!=	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน	`echo $x != $z;`	false
<>	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน	`echo $x <> $y;`	true
!==	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน หรือ ประเภทตัวแปรไม่เหมือนกัน	`echo $x !== $z;`	true
>	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่ามากกว่า	`echo $x > $y;`	true
<	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่าน้อยกว่า	`echo $x < $y;`	false
>=	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่ามากกว่า หรือเท่ากับ	`echo $x >= $y;`	true
<=	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่าน้อยกว่า หรือเท่ากับ	`echo $x <= $y;`	false

ตัวดำเนินการเพิ่มค่าหรือลดค่า (Increment/Decrement operators)

ตัวดำเนินการเพิ่มค่าหรือลดค่า ใช้เพิ่มค่าหรือลดค่าตัวแปรประเภทตัวเลขทีละ 1 ค่า

ตัวดำเนินการ	ชื่อ	ตัวอย่าง กำหนดให้ `$x = 5;`	ความหมาย	ผลลัพธ์
`++$x`	Pre-increment	`echo ++$x;`	เพิ่มค่าออก 1 แล้วค่อยแสดงผล	6
`--$x`	Post-increment	`echo --$x;`	ลบค่าออก 1 แล้วค่อยแสดงผล	4
`$x++`	Pre-decrement	`echo $x++;`	แสดงผลแล้วค่อยเพิ่มค่า 1 ให้กับตัวแปร `$x`	5
`$x--`	Post-decrement	`echo $x--;`	แสดงผลแล้วค่อยลบค่า 1 ออกจากตัวแปร `$x`	5

ตัวดำเนินการเชิงตรรกะ (Logical operators)

ตัวดำเนินการเชิงตรรกะ ใช้กับคำสั่งที่เป็นเงื่อนไขเพื่อตรวจสอบว่าเป็นจริงหรือเป็นเท็จ

ตัวดำเนินการ	ชื่อ	ตัวอย่าง กำหนดให้ `$x = true;` `$y = false;`	ความหมาย	ผลลัพธ์
`and`	และ	`$x and $y;`	จะเป็นจริงก็ต่อเมื่อ $x,$y เป็นจริงทั้งสอง	`false`
`or`	หรือ	`$x or $y;`	จะเป็นจริงก็ต่อเมื่อ $x,$y เป็นจริงตัวใดตัวหนึ่ง	`true`
`xor`	xor	`$x xor $y;`	จะเป็นจริงก็ต่อเมื่อ $x,$y เป็นจริงตัวใดตัวหนึ่ง และห้ามเป็นจริงทั้ง 2 ตัว	`true`
`&&`	และ	`$x && $y;`	จะเป็นจริงก็ต่อเมื่อ $x,$y เป็นจริงทั้งสอง	`false`
`||`	หรือ	`$x || $y;`	จะเป็นจริงก็ต่อเมื่อ $x,$y เป็นจริงตัวใดตัวหนึ่ง	`true`
`!`	ไม่ใช่	`!$x;`	จะเป็นจริงก็ต่อเมื่อ $x เป็นเท็จ	`false`

ตัวดำเนินการเชิงข้อความ (String operators)

ตัวดำเนินการเชิงข้อความ ใช้งานโดยการเอาข้อความสองข้อความเรียงต่อกัน มี 2 รูปแบบ ดังนี้

เครื่องหมายจุด (.) คือ การนำข้อความทั้ง 2 ข้อความมาเรียงต่อกันตัวอย่างเช่น

```
<?php
    $x = "Chala";
```

```php
    $y = "wan";
    $z = $x.$y; //นำข้อความจากตัวแปร xและ y มาเรียงต่อกัน
    echo $z; //ผลลัพธ์ Chalawan
?>
```

เครื่องหมายจุดเท่ากับ (**.=**) คือ การนำตัวแปร 2 ตัว มาเรียงต่อกัน โดยให้เก็บค่าลงที่ตัวแปรที่อยู่ด้านซ้าย เช่น $x .= $y (นำข้อความจากตัวแปร y มาเรียงต่อกับตัวแปร x และเก็บค่าไว้ที่ตัวแปร x)

ตัวอย่าง

```php
<?php
    $x = "Chala";
    $y = "wan";
    $x .= $y; //นำข้อความจากตัวแปร xและ y มาเรียงต่อกันและเก็บค่าที่ได้ไว้ที่ตัวแปร x
    echo $x; //ผลลัพธ์ Chalawan
?>
```

ตัวดำเนินการเชิงอาร์เรย์ (Array operators)

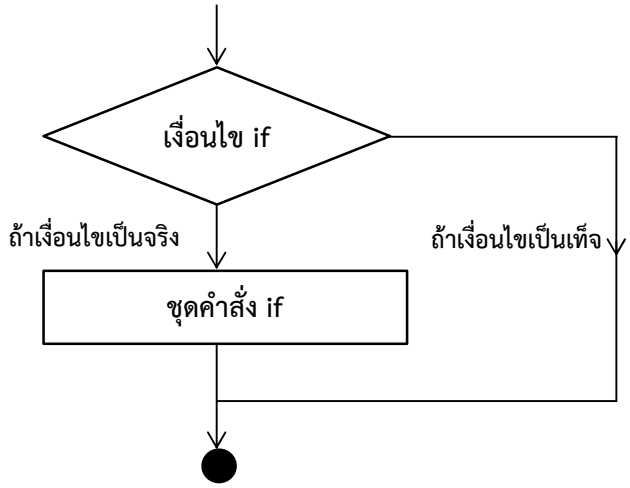
ใช้โดยการนำตัวแปรอาร์เรย์ 2 ตัวมาเปรียบเทียบกันโดยใช้เครื่องหมายดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
กำหนดให้ `$x = array("a" => "red", "b" => "green");` `$y = array("c" => "blue", "d" => "yellow");`			
+	นำค่าของ 2 ตัวแปรอาร์เรย์รวมกัน	`print_r($x + $y);`	ค่าทั้งหมดของตัวแปร x และ y
==	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าเท่ากัน	`var_dump($x == $y);`	**false**
===	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าเท่ากัน **และ** ประเภทตัวแปรเหมือนกัน	`var_dump($x === $y);`	**false**
!=	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน	`var_dump($x != $y);`	**true**
<>	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน	`var_dump($x <> $y);`	**true**
!==	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน หรือ ประเภทตัวแปรไม่เหมือนกัน	`var_dump($x !== $y);`	**true**

เงื่อนไข if else elseif switch

การเขียนคำสั่งเงื่อนไขโดยใช้ if else elseif switch เป็นการเขียนโค๊ดเพื่อให้โปรแกรมทำงานตามความต้องการ ถ้าเงื่อนไขนั้นเป็นจริง (True) และ ถ้าทำในสิ่งที่นอกเหนือจากความต้องการ ในกรณีเงื่อนไขเป็นเท็จ (False) ซึ่งสามารถแบ่งหัวข้อของการใช้เงื่อนไขดังต่อไปนี้

1. เงื่อนไขทางเลือกเดียว (if) เป็นการกำหนดเงื่อนไขทิศทางเดียว โดยที่โปรแกรมจะทำงานตามเงื่อนไขในกรณีที่เงื่อนไขเป็นจริงเท่านั้นซึ่งมีรูปแบบดังนี้



ภาพที่ 2 เงื่อนไขแบบทางเลือกเดียว

รูปแบบคำสั่ง

```
if (เงื่อนไข) {
    สิ่งที่ต้องการให้ทำถ้าเงื่อนไขเป็นจริง หรือ ชุดคำสั่ง;
}
```

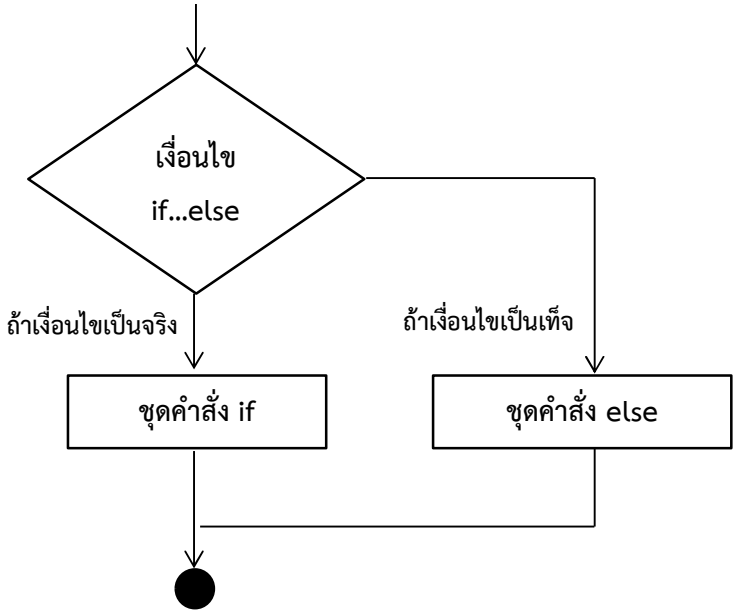
ตัวอย่าง

```php
<?php
    $score = 60;
    if($score >= 50){
        echo "ผ่าน";
    }
?>
```

ผลลัพธ์

ผ่าน

2. เงื่อนไขแบบสองทางเลือก (if...else) เป็นการกำหนดเงื่อนไขสองทิศทาง โดยที่โปรแกรมจะทำงานตามความต้องการ เมื่อเงื่อนไขเป็นจริง (True) โปรแกรมจะทำงานตามคำสั่งในส่วนของ if และถ้าเงื่อนไขเป็นเท็จ (False) โปรแกรมจะทำงานตามคำสั่งในส่วนของ else ซึ่งมีรูปแบบดังนี้



ภาพที่ 3 เงื่อนไขแบบสองทางเลือก if...else

รูปแบบคำสั่ง

```
if (เงื่อนไข) {
  สิ่งที่ต้องการให้ทำถ้าเงื่อนไขเป็นจริง หรือ ชุดคำสั่ง if;
}else{
  สิ่งที่ต้องการให้ทำถ้าเงื่อนไขเป็นเท็จ หรือ ชุดคำสั่ง else;
}
```

ตัวอย่าง

```php
<?php
    $score = 40;
    if($score >= 50){
        echo "ผ่าน";
    }else{
        echo "ไม่ผ่าน";
    }
?>
```

ผลลัพธ์

ไม่ผ่าน

แต่ในบางกรณีการใช้ if...else ไม่เพียงพอต่อการทำงาน และต้องการจะเพิ่มเงื่อนไขให้ทำงานหลายขั้นตอนมากขึ้น หรือต้องการกำหนดเงื่อนไขหลายๆ มากกว่า 2 เงื่อนไข สามารถ ใช้คำสั่ง elseif ได้ดังนี้

รูปแบบคำสั่ง

```
if (เงื่อนไข) {
  สิ่งที่ต้องการให้ทำถ้าเงื่อนไขเป็นจริง หรือ ชุดคำสั่ง if;
}elseif (เงื่อนไข) {
     สิ่งที่ต้องการให้ทำถ้าเงื่อนไขเป็นจริง หรือ ชุดคำสั่ง elseif;
}else{
     สิ่งที่ต้องการให้ทำถ้าเงื่อนไขเป็นเท็จ หรือ ชุดคำสั่ง else;
}
```

ตัวอย่าง

```php
<?php
    $score = 70;
    if($score>=80){
        echo "ดีมาก";
    }elseif($score>=60){
        echo "ปานกลาง";
    }else{
        echo "น้อยมาก";
    }
?>
```

ผลลัพธ์

ปานกลาง

3. Switch เป็นเงื่อนไขใช้สำหรับดำเนินการต่างๆ ตามเงื่อนไขที่ต่างกัน โดยการทำงานจะเจาะจงไปยังส่วนที่ต้องการจะทำงานทันที ซึ่งแตกต่างจากเงื่อนไข if else elseif ที่ทำงานเปรียบเทียบทีละบรรทัดเมื่อเงื่อนไขเป็นจริงถึงจะประมวลผลชุดคำสั่งในส่วนนั้น รูปแบบเงื่อนไข switch มีการทำงานดังนี้

รูปแบบคำสั่ง

```
switch (n) {
    case label1:
        code to be executed if n=label1;
        break;
    case label2:
        code to be executed if n=label2;
        break;
    case label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

ตัวอย่าง

```php
<?php
    $favcolor = "red";
    switch ($favcolor) {
        case "red":
            echo "สีที่คุณชอบ คือ สีแดง";
            break;
        case "blue":
            echo "สีที่คุณชอบ คือ สีฟ้า";
            break;
        case "green":
            echo "สีที่คุณชอบ คือ สีเขียว";
            break;
        default:
            echo "สีที่คุณชอบนอกเหลือจาก สีแดง สีฟ้า และ สีเขียว";
    }
?>
```

ผลลัพธ์

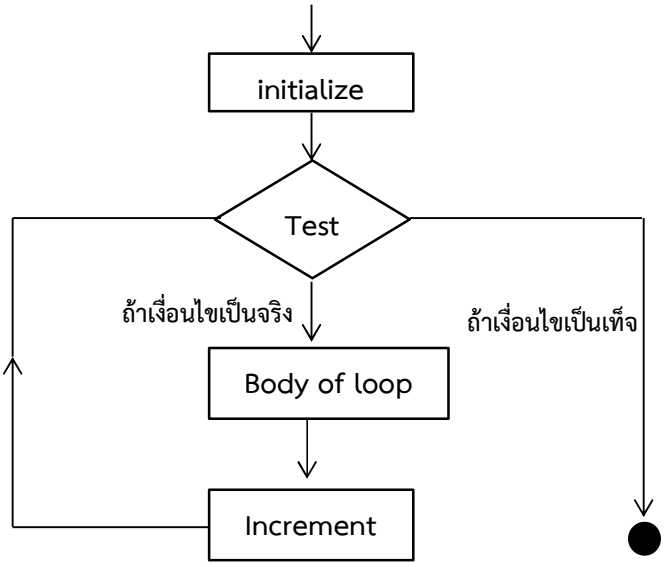
สีที่คุณชอบ คือ สีแดง

คำสั่งทำซ้ำ (Loop)

คำสั่งทำซ้ำ (Loop) คือ คำสั่งที่ทำงานซ้ำๆ วนไปมา เมื่อเงื่อนไขเป็นจริง แต่เมื่อไหร่ที่เงื่อนไขเป็นเท็จ คำสั่งทำซ้ำ จะจบการทำงานทันที ซึ่งสามารถแบ่งออกเป็น 2 คำสั่ง คือ For และ While

1. For Loop

ใช้ For Loop เมื่อทราบว่าต้องการให้โปรแกรมทำงานซ้ำๆ จำนวนกี่รอบ ดังรูปดังต่อไปนี้



ภาพที่ 4 โครงสร้างการทำงานของลูป For

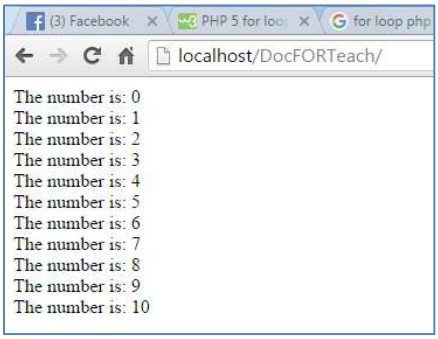
รูปแบบคำสั่ง

for (*เริ่มต้นการนับ; ทดสอบโค๊ด; เพิ่มค่าให้กับการนับ*) {
โค๊ดจะทำงานเมื่อคำสั่ง for เป็นจริง
}

ตัวอย่างที่ 1 จงแสดงเลข 0-10

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

ผลลัพธ์ 1



ภาพที่ 5 ผลลัพธ์การทำงานของลูป For

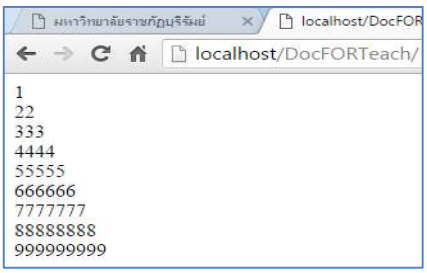
ตัวอย่างที่ 2 จงแสดงจำนวนเต็มที่เป็นเลขคี่ตั้งแต่เลข 1-10

```php
<?PHP
    for ($x = 1;$x <= 10;$x++){
        if($x % 2 == 1){
            echo $x;
        }
    }
?>
```

ผลลัพธ์ที่ 2

13579

ตัวอย่างที่ 3 จงเขียนโค๊ดให้แสดงผลลัพธ์ดังต่อไปนี้



ภาพที่ 6 ผลลัพธ์ของตัวอย่างที่ 3

คำตอบตัวอย่างที่ 3

```php
<?PHP
    for ($x = 1;$x <= 9;$x++){
        for ($y = 1 ;$y <= $x; $y++){
            echo $x;
        }
        echo "<br>";
    }
?>
```

2. Foreach Loop

Foreach Loop เป็นคำสั่งที่ใช้ทำงานกับตัวแปรที่เป็น Array ใช้การวนซ้ำไปเรื่อยๆ ตามจำนวนสมาชิกใน Array นั้นๆ

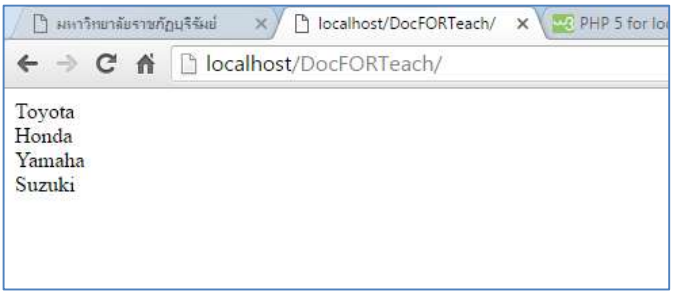
รูปแบบคำสั่ง

```
foreach ($array as $value) {
    คำสั่งที่จะทำงานเมื่อเงื่อนไข foreach เป็นจริง
}
```

ตัวอย่าง

```php
<?php
    $cars = array("Toyota", "Honda", "Yamaha", "Suzuki");
    foreach ($cars as $value) {
        echo "$value <br>";
    }
?>
```

ผลลัพธ์



ภาพที่ 7 ผลลัพธ์ของ foreach loop

3. While Loop

While Loop จะดำเนินการทำงานกับโค๊ดภายใต้คำสั่ง while loop เมื่อเงื่อนไขเป็นจริงเท่านั้น

รูปแบบคำสั่ง

```
while (เงื่อนไขเป็นจริง) {
    คำสั่งที่จะทำงานเมื่อเงื่อนไข while เป็นจริง
}
```

ตัวอย่างที่ 1

```php
<?php
    $x = 1;
    while($x <= 5) {
        echo "$x";
        $x++;
    }
?>
```

ผลลัพธ์ที่ 1

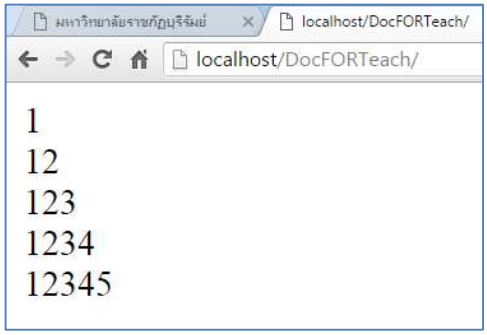
12345

คำสั่ง while loop สามารถเขียนคำสั่งซ้อนกันได้ตามความต้องการของผู้เขียนโค๊ดดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2

```php
<?php
    $x = 1;
    while($x <= 5) {
        $y = 1;
        while ($y<=$x){
            echo "$y";
            $y++;
        }
        echo "<br>";
        $x++;
    }
?>
```

ผลลัพธ์ที่ 2



ภาพที่ 8 ผลลัพธ์ของ while loop

ฟังก์ชัน (Function)

ฟังก์ชันการทำงานในภาษา php มีหลายฟังก์ชัน สามารถเรียกใช้ฟังก์ชันไหนก็ได้ตามที่ต้องการให้เหมาะสมกับงานที่ทำ แต่บางครั้งฟังก์ชันในภาษา php ไม่สามารถทำงานได้ตามที่ต้องการทั้งหมด จึงจำเป็นต้องเขียนฟังก์ชันขึ้นมาด้วยตนเองซึ่งมีข้อกำหนดดังนี้

1. ฟังก์ชันเป็นชุดคำสั่งที่สามารถใช้ซ้ำๆ ในโปรแกรมได้
2. ฟังก์ชันจะไม่ทำงานในทันทีเมื่อโหลดหน้าเว็บ
3. ฟังก์ชันจะทำงานก็ต่อเมื่อมีการเรียกใช้ฟังก์ชัน
4. การตั้งชื่อฟังก์ชันจะต้องไม่ตรงกับชื่อฟังก์ชันที่มีอยู่ภาษา php
5. ขึ้นต้นด้วยตัวอักษรภาษาอังกฤษ
6. ห้ามเว้นวรรคระหว่างตั้งชื่อฟังก์ชัน

รูปแบบของฟังก์ชัน

```
function ชื่อฟังก์ชัน() {
    โค๊ดที่ต้องการให้ฟังก์ชันทำงาน
}
```

ตัวอย่างที่ 1

```php
<?php
    function Message() {
        echo "Hello world";
    }
    Message(); // การเรียกใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 1

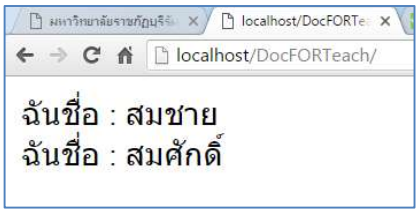
Hello world

จากตัวอย่างที่ 1 เป็นฟังก์ชันที่แสดงผลตามที่ต้องการให้แสดงเท่านั้น นอกจากนี้ยังสามารถส่งข้อมูลเข้าไปในฟังก์ชันได้ โดยผ่านอาร์กิวเมนต์ของฟังก์ชัน ซึ่งจะกำหนดอาร์กิวเมนต์ไว้ในวงเล็บของฟังก์ชัน และสามารถกำหนดได้มากกว่า 1 อาร์กิวเมนต์ โดยขั้นด้วยเครื่องหมายจุลภาค (,) แสดงดังตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 2

```php
<?php
    echo '<meta charset="utf-8">';
    function name($fname) {
        echo "ฉันชื่อ: $fname <br>";
    }
    name("สมชาย");//การเขียนใช้ฟังก์ชัน
    name("สมศักดิ์");//การเขียนใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 2

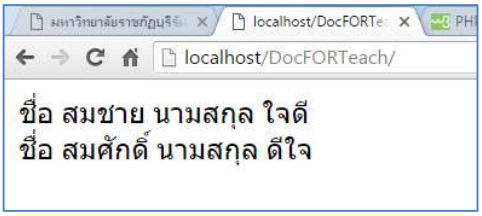


ภาพที่ 9 ผลลัพธ์ของฟังก์ชันอาร์กิวเมนต์

ตัวอย่างที่ 3

```php
<?php
    echo '<meta charset="utf-8">';
    function name($fname,$lname) {
        echo "ชื่อ $fname นามสกุล $lname <br>";
    }
    name("สมชาย","ใจดี"); //การเขียนใช้ฟังก์ชัน
    name("สมศักดิ์","ดีใจ");//การเขียนใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 3



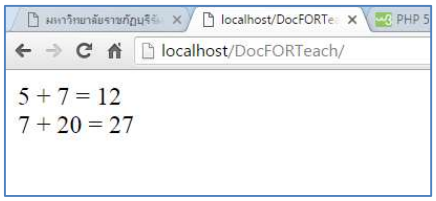
ภาพที่ 10 ผลลัพธ์ของฟังก์ชันอาร์กิวเมนต์มากกว่า 1 ตัว

นอกจากนี้ฟังก์ชันยังสามารถส่งค่ากลับ (return) เพื่อนำค่าที่ได้ไปคำนวณทำอย่างอื่นได้อีก เช่น นำค่าไปคำนวณทางคณิตศาสตร์ เป็นต้น เรียกว่า ฟังก์ชันรีเทิร์น แสดงดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 4

```php
<?php
    function sum($x, $y) {
        $z = $x + $y;
        return $z;
    }
    echo "5 + 7 = " . sum(5, 7) . "<br>"; //การเรียกใช้ฟังก์ชัน
    echo "7 + 20 = " . sum(7, 20);   //การเรียกใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 4



ภาพที่ 11 ผลลัพธ์ของฟังก์ชันรีเทิร์น

ตัวแปรหลายค่า (Array)

ตัวแปรอาร์เรย์ เป็นตัวแปรที่สามารถเก็บค่าไว้ได้หลายค่าในตัวแปรเดียว ซึ่งมีรูปแบบการเขียนดังนี้

$val = array(“แดง”,20) ;

$val หมายถึง ชื่อตัวแปร

“แดง” หมายถึง ค่าที่เป็นตัวอักษร

20 หมายถึง ค่าที่เป็นตัวเลข

การใช้งานตัวแปร array ให้เรียกตามตำแหน่ง index ของค่านั้นๆ ซึ่งจะเริ่มจาก 0 ดังตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 1

```php
<?php
    echo '<meta charset="utf-8">';
    $col = array("แดง", "ม่วง", "เขียว");
    echo "สีที่ฉันชอบคือ " . $col[0] . " " . $col[1] . " และ " . $col[2] ;
?>
```

ผลลัพธ์ที่ 1

สีที่ฉันชอบคือ แดง ม่วง และ เขียว

ถ้าหากต้องการเปลี่ยนค่าของตัวแปรอาร์เรย์ให้อ้างไปที่ index ของค่านั้นๆ เช่น

ตัวอย่างที่ 2

```php
<?php
    echo '<meta charset="utf-8">';
    $col = array("แดง", "ม่วง", "เขียว");
    $col[0] = "ดำ";
    $col[1] = "ขาว";
    $col[2] = "น้ำเงิน";
    echo "สีที่ฉันชอบคือ " . $col[0] . " " .  $col[1] . " และ " . $col[2];
?>
```

ผลลัพธ์ที่ 2

สีที่ฉันชอบคือ ดำ ขาว และ น้ำเงิน

การนับจำนวนสมาชิก array โดยใช้ฟังก์ชัน count( )

```php
<?php
    $x = array("A", "B", "C");
    echo count($x);
?>
```

ผลลัพธ์

3

ตัวอย่าง

```php
<?php
    $favcolor = "red";
    switch ($favcolor) {
        case "red":
            echo "สีที่คุณชอบ คือ สีแดง";
            break;
        case "blue":
            echo "สีที่คุณชอบ คือ สีฟ้า";
            break;
        case "green":
            echo "สีที่คุณชอบ คือ สีเขียว";
            break;
        default:
            echo "สีที่คุณชอบนอกเหลือจาก สีแดง สีฟ้า และ สีเขียว";
    }
?>
```

ผลลัพธ์

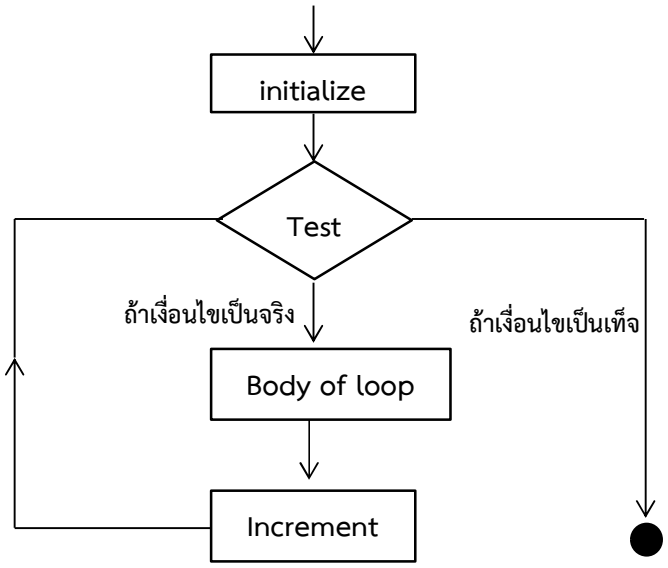
สีที่คุณชอบ คือ สีแดง

คำสั่งทำซ้ำ (Loop)

คำสั่งทำซ้ำ (Loop) คือ คำสั่งที่ทำงานซ้ำๆ วนไปมา เมื่อเงื่อนไขเป็นจริง แต่เมื่อไหร่ที่เงื่อนไขเป็นเท็จ คำสั่งทำซ้ำ จะจบการทำงานทันที ซึ่งสามารถแบ่งออกเป็น 2 คำสั่ง คือ For และ While

1. For Loop

ใช้ For Loop เมื่อทราบว่าต้องการให้โปรแกรมทำงานซ้ำๆ จำนวนกี่รอบ ดังรูปดังต่อไปนี้



ภาพที่ 4 โครงสร้างการทำงานของลูป For

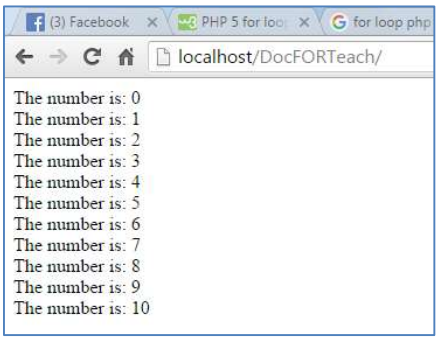
รูปแบบคำสั่ง

for (*เริ่มต้นการนับ; ทดสอบโค๊ด; เพิ่มค่าให้กับการนับ*) {
โค๊ดจะทำงานเมื่อคำสั่ง for เป็นจริง
}

ตัวอย่างที่ 1 จงแสดงเลข 0-10

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

ผลลัพธ์ 1



ภาพที่ 5 ผลลัพธ์การทำงานของลูป For

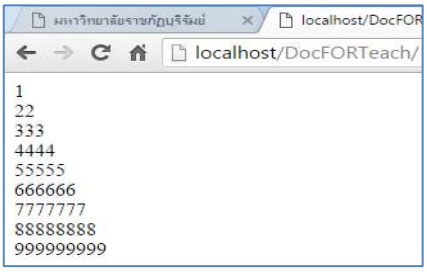
ตัวอย่างที่ 2 จงแสดงจำนวนเต็มที่เป็นเลขคี่ตั้งแต่เลข 1-10

```php
<?PHP
    for ($x = 1;$x <= 10;$x++){
        if($x % 2 == 1){
            echo $x;
        }
    }
?>
```

ผลลัพธ์ที่ 2

13579

ตัวอย่างที่ 3 จงเขียนโค๊ดให้แสดงผลลัพธ์ดังต่อไปนี้



ภาพที่ 6 ผลลัพธ์ของตัวอย่างที่ 3

คำตอบตัวอย่างที่ 3

```php
<?PHP
    for ($x = 1;$x <= 9;$x++){
        for ($y = 1 ;$y <= $x; $y++){
            echo $x;
        }
        echo "<br>";
    }
?>
```

2. Foreach Loop

Foreach Loop เป็นคำสั่งที่ใช้ทำงานกับตัวแปรที่เป็น Array ใช้การวนซ้ำไปเรื่อยๆ ตามจำนวนสมาชิกใน Array นั้นๆ

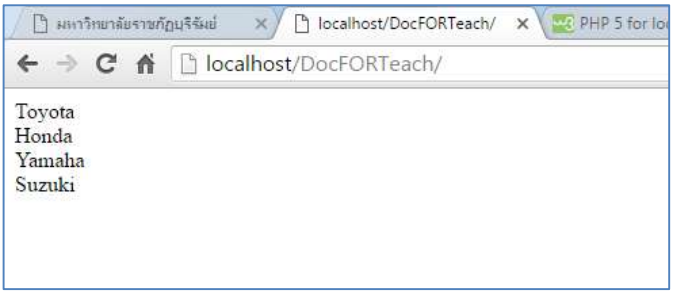
รูปแบบคำสั่ง

```
foreach ($array as $value) {
    คำสั่งที่จะทำงานเมื่อเงื่อนไข foreach เป็นจริง
}
```

ตัวอย่าง

```php
<?php
    $cars = array("Toyota", "Honda", "Yamaha", "Suzuki");
    foreach ($cars as $value) {
        echo "$value <br>";
    }
?>
```

ผลลัพธ์



ภาพที่ 7 ผลลัพธ์ของ foreach loop

3. While Loop

While Loop จะดำเนินการทำงานกับโค๊ดภายใต้คำสั่ง while loop เมื่อเงื่อนไขเป็นจริงเท่านั้น

รูปแบบคำสั่ง

```
while (เงื่อนไขเป็นจริง) {
    คำสั่งที่จะทำงานเมื่อเงื่อนไข while เป็นจริง
}
```

ตัวอย่างที่ 1

```php
<?php
    $x = 1;
    while($x <= 5) {
        echo "$x";
        $x++;
    }
?>
```

ผลลัพธ์ที่ 1

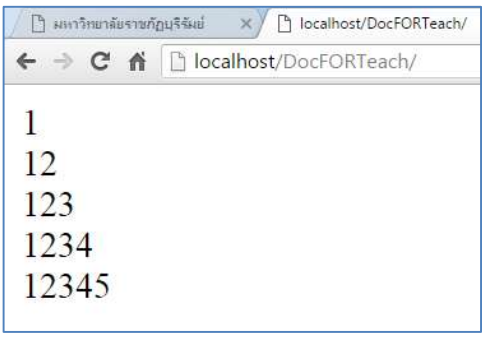
12345

คำสั่ง while loop สามารถเขียนคำสั่งซ้อนกันได้ตามความต้องการของผู้เขียนโค๊ดดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2

```php
<?php
    $x = 1;
    while($x <= 5) {
        $y = 1;
        while ($y<=$x){
            echo "$y";
            $y++;
        }
        echo "<br>";
        $x++;
    }
?>
```

ผลลัพธ์ที่ 2



ภาพที่ 8 ผลลัพธ์ของ while loop

ฟังก์ชัน (Function)

ฟังก์ชันการทำงานในภาษา php มีหลายฟังก์ชัน สามารถเรียกใช้ฟังก์ชันไหนก็ได้ตามที่ต้องการให้เหมาะสมกับงานที่ทำ แต่บางครั้งฟังก์ชันในภาษา php ไม่สามารถทำงานได้ตามที่ต้องการทั้งหมด จึงจำเป็นต้องเขียนฟังก์ชันขึ้นมาด้วยตนเองซึ่งมีข้อกำหนดดังนี้

1. ฟังก์ชันเป็นชุดคำสั่งที่สามารถใช้ซ้ำๆ ในโปรแกรมได้

2. ฟังก์ชันจะไม่ทำงานในทันทีเมื่อโหลดหน้าเว็บ
3. ฟังก์ชันจะทำงานก็ต่อเมื่อมีการเรียกใช้ฟังก์ชัน
4. การตั้งชื่อฟังก์ชันจะต้องไม่ตรงกับชื่อฟังก์ชันที่มีอยู่ภาษา php
5. ขึ้นต้นด้วยตัวอักษรภาษาอังกฤษ
6. ห้ามเว้นวรรคระหว่างตั้งชื่อฟังก์ชัน

รูปแบบของฟังก์ชัน

```
function ชื่อฟังก์ชัน() {
    โค๊ดที่ต้องการให้ฟังก์ชันทำงาน
}
```

ตัวอย่างที่ 1

```php
<?php
    function Message() {
        echo "Hello world";
    }
    Message(); // การเรียกใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 1

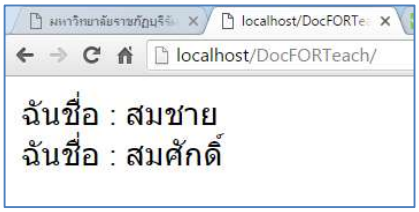
Hello world

จากตัวอย่างที่ 1 เป็นฟังก์ชันที่แสดงผลตามที่ต้องการให้แสดงเท่านั้น นอกจากนี้ยังสามารถส่งข้อมูลเข้าไปในฟังก์ชันได้ โดยผ่านอาร์กิวเมนต์ของฟังก์ชัน ซึ่งจะกำหนดอาร์กิวเมนต์ไว้ในวงเล็บของฟังก์ชัน และสามารถกำหนดได้มากกว่า 1 อาร์กิวเมนต์ โดยขั้นด้วยเครื่องหมายจุลภาค (,) แสดงดังตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 2

```php
<?php
    echo '<meta charset="utf-8">';
    function name($fname) {
        echo "ฉันชื่อ: $fname <br>";
    }
    name("สมชาย");//การเขียนใช้ฟังก์ชัน
    name("สมศักดิ์");//การเขียนใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 2



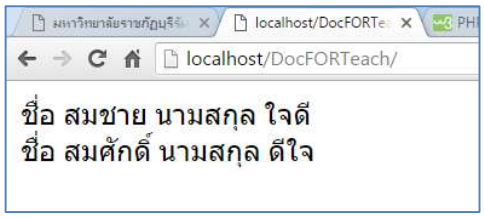
ภาพที่ 9 ผลลัพธ์ของฟังก์ชันอาร์กิวเมนต์

ตัวอย่างที่ 3

```php
<?php
```

```
    echo '<meta charset="utf-8">';
    function name($fname,$lname) {
        echo "ชื่อ  $fname นามสกุล  $lname <br>";
    }
    name("สมชาย","ใจดี"); //การเขียนใช้ฟังก์ชัน
    name("สมศักดิ์","ดีใจ");//การเขียนใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 3



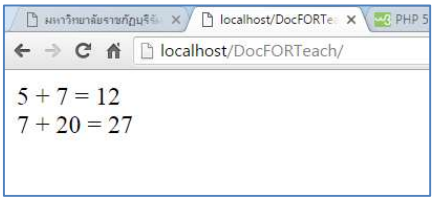
ภาพที่ 10 ผลลัพธ์ของฟังก์ชันอาร์กิวเมนต์มากกว่า 1 ตัว

นอกจากนี้ฟังก์ชันยังสามารถส่งค่ากลับ (return) เพื่อนำค่าที่ได้ไปคำนวณทำอย่างอื่นได้อีก เช่น นำค่าไปคำนวณทางคณิตศาสตร์ เป็นต้น เรียกว่า ฟังก์ชันรีเทิร์น แสดงดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 4

```
<?php
    function sum($x, $y) {
        $z = $x + $y;
        return $z;
    }
    echo "5 + 7 = " . sum(5, 7) . "<br>"; //การเรียกใช้ฟังก์ชัน
    echo "7 + 20 = " . sum(7, 20);   //การเรียกใช้ฟังก์ชัน
?>
```

ผลลัพธ์ที่ 4



ภาพที่ 11 ผลลัพธ์ของฟังก์ชันรีเทิร์น

ตัวแปรหลายค่า (Array)

ตัวแปรอาร์เรย์ เป็นตัวแปรที่สามารถเก็บค่าไว้ได้หลายค่าในตัวแปรเดียว ซึ่งมีรูปแบบการเขียนดังนี้

$val = array(“แดง”,20) ;

$val หมายถึง ชื่อตัวแปร

“แดง” หมายถึง ค่าที่เป็นตัวอักษร

20 หมายถึง ค่าที่เป็นตัวเลข

การใช้งานตัวแปร array ให้เรียกตามตำแหน่ง index ของค่านั้นๆ ซึ่งจะเริ่มจาก 0 ดังตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 1

```
<?php
    echo '<meta charset="utf-8">';
```

```
    $col = array("แดง", "ม่วง", "เขียว");
    echo "สีที่ฉันชอบคือ " . $col[0] . " " . $col[1] . " และ " . $col[2] ;
?>
```

ผลลัพธ์ที่ 1

สีที่ฉันชอบคือ แดง ม่วง และ เขียว

ถ้าหากต้องการเปลี่ยนค่าของตัวแปรอาร์เรย์ให้อ้างไปที่ index ของค่านั้นๆ เช่น

ตัวอย่างที่ 2

```
<?php
    echo '<meta charset="utf-8">';
    $col = array("แดง", "ม่วง", "เขียว");
    $col[0] = "ดำ";
    $col[1] = "ขาว";
    $col[2] = "น้ำเงิน";
    echo "สีที่ฉันชอบคือ " . $col[0] . " " .  $col[1] . " และ " . $col[2];
?>
```

ผลลัพธ์ที่ 2

สีที่ฉันชอบคือ ดำ ขาว และ น้ำเงิน

การนับจำนวนสมาชิก array โดยใช้ฟังก์ชัน count( )

```
<?php
    $x = array("A", "B", "C");
    echo count($x);
?>
```

ผลลัพธ์

3

PHP กับ MySQL

ภาษา html ซึ่งเป็นภาษาหลักในการแสดงหน้าเว็บ ไม่สามารถติดต่อกับฐานข้อมูลได้ ดังนั้น จำเป็นต้องใช้ภาษาอื่นที่สามารถติดต่อไปยังฐานข้อมูลได้ ซึ่งภาษา php เป็นภาษาที่สามารถติดต่อกับฐานข้อมูลได้ และเป็นที่รู้จักกันอย่างแพร่หลายในวงการนักพัฒนาโปรแกรม มีวิธีการดังนี้

เชื่อมต่อกับ MySQL

```
<meta charset="utf-8">
<?php
$servername = "localhost";
$username = "root";
$password = "root";

// Create connection
$conn = mysqli_connect($servername, $username, $password);
mysqli_set_charset($conn, "utf8");

// Check connection
if (!$conn) {
    echo "เชื่อมต่อไม่ได้ NOTTTTTT<BR>";
}else{
    echo "เชื่อมต่อได้ OKKKKK <BR>";
}
mysqli_close($conn);
?>
```

สร้างฐานข้อมูล

```
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "สร้างฐานข้อมูลเรียบร้อย OKKKK";
} else {
    echo "ผิดพลาด NOTTTTTTT: " . mysqli_error($conn);
}
```

สร้างตาราง

```
$dbname = "myDB"; //เพิ่มต่อจากตัวแปร $password
$conn = mysqli_connect($servername, $username, $password, $dbname);
$sql = "CREATE TABLE tourist (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name_tourist VARCHAR(30) NOT NULL,
lat double(10,6) DEFAULT NULL,
lon double(10,6) DEFAULT NULL,
date date DEFAULT NULL
)ENGINE=MyISAM DEFAULT CHARSET=tis620;";

if (mysqli_query($conn, $sql)) {
    echo "สร้างตารางเรียบร้อย";
} else {
    echo "ผิดพลาด:" . mysqli_error($conn);
}
```

เพิ่มข้อมูล(insert)

```
$sql = "INSERT INTO tourist (name_tourist, lat, lon,date)
VALUES ('เขากระโดง', '14.939530', ' 103.092004',NOW())";

if (mysqli_query($conn, $sql)) {
    echo "บันทึกเรียบร้อย";
} else {
    echo "ผิดพลาด:" . $sql . "<br>" . $conn->error;
}
```

เพิ่มข้อมูล และ รับค่า id ตัวสุดท้ายมาด้วย (insert)

```
$sql = "INSERT INTO tourist (name_tourist, lat, lon,date)
VALUES ('เขากระโดง', '14.939530', ' 103.092004',NOW())";

if (mysqli_query($conn, $sql)) {
    $last_id = mysqli_insert_id($conn);
    echo "บันทึกข้อมูลเรียบร้อย. id ล่าสุด คือ :" . $last_id;
} else {
    echo "ผิดพลาด <BR>: " . $sql . "<br>" . mysqli_error($conn);
}
```

แสดงข้อมูล (select)

```
$sql = "SELECT * FROM tourist";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
```

```
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " Name: " . $row["name_tourist"]. "
" . $row["lat"]. " " . $row["lon"]. "<br>";
    }
} else {
    echo "0 results";
}
```

ลบข้อมูล

```
$sql = "DELETE FROM tourist WHERE id=1";

if (mysqli_query($conn, $sql)) {
    echo "ลบข้อมูลเรียบร้อย";
} else {
    echo "ผิดพลาด:" . mysqli_error($conn);
}
```

แก้ไขข้อมูล

```
$sql = "UPDATE tourist SET name_tourist='สวนนก' WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "แก้ไขข้อมูลเรียบร้อย";
} else {
    echo "ผิดพลาด:" . mysqli_error($conn);
}
```

แสดงข้อมูลแบบจำกัด

```
$sql = "SELECT * FROM tourist LIMIT 3";
```
แสดงแค่ 3 แถวเท่านั้น

```
$sql = "SELECT * FROM tourist LIMIT 3 OFFSET 2";
```
ไม่แสดง 2 แถวแรก และ แสดงแถวที่3 4 5

```
$sql = "SELECT * FROM tourist LIMIT 1,3";
```
ไม่แสดงแถวที่ 1 และ แสดงแถวที่ 2 3 4

บทสรุป

PHP ย่อมาจาก PHP Hypertext Preprocessor แต่เดิมย่อมาจาก Personal Home Page Tools PHP คือภาษาคอมพิวเตอร์จำพวก scripting language ภาษาจำพวกนี้คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า Script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ตัวอย่างของภาษาสคริปก็เช่น JavaScript , Perl เป็นต้น เกิดในปี 1994 โดย Rasmus Lerdorf โปรแกรมเมอร์อเมริกันได้คิดค้นสร้างเครื่องมือที่ใช้ในการพัฒนาเว็บส่วนตัวของเขา โดยใช้ข้อดีของภาษา C และ Perl เรียกว่า Personal Home Page และได้สร้างส่วนติดต่อกับฐานข้อมูลที่ชื่อว่า Form Interpreter (FI) รวมทั้งสองส่วน เรียกว่า PHP/FI

โปรแกรมที่พัฒนาขึ้นโดยภาษา PHP จำเป็นต้องติดตั้งเว็บเซิร์ฟเวอร์บนพีซีจากนั้นติดตั้ง PHP และ MySQL เพื่อให้ code สามารถทำงานได้บนเว็บบราวเซอร์

คำถามท้ายบทที่ 3

1. จงอธิบายลักษณะการทำงานของภาษา PHP
2. จงอธิบายการสร้างตัวแปร และการทำงานของตัวแปร พร้อมทั้งยกตัวอย่างประกอบ
3. จงเขียนโค๊ดแสดงข้อความดังนี้ “Hello World”
4. จงเขียนโครงสร้างการทำงานของ if else พร้อมอธิบาย และยกตัวอย่างโค๊ดประกอบ
5. จงอธิบายความแตกต่างระหว่าง loop for กับ loop while พร้อมทั้งบอกประโยชน์

เอกสารอ้างอิง

พีเอชพี. (2564). **พีเอชพีคืออะไร.** ค้นเมื่อ 13 ตุลาคม 2564, จาก http://www.mindphp.com/คู่มือ/73-คืออะไร/2127-php-คืออะไร.html

w3schools. (2564). **PHP Tutorial.** ค้นเมื่อ 14 ตุลาคม 2564, จาก https://www.w3schools.com/php/default.asp