

## ภาษา PHP

PHP ย่อมาจาก PHP Hypertext Preprocessor แต่เดิมนย่อมาจาก Personal Home Page Tools PHP คือภาษาคอมพิวเตอร์จำพวก scripting language ภาษาจำพวกนี้คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า Script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ตัวอย่างของภาษาสคริปก็เช่น JavaScript , Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวได้ว่า PHP เป็นภาษาที่เรียกว่า Server-side หรือ HTML-embedded scripting language นั่นคือในทุกๆ ครั้งก่อนที่เครื่องคอมพิวเตอร์ซึ่งให้บริการเป็น Web server จะส่งหน้าเว็บเพจที่เขียนด้วย PHP ให้เรา มันจะทำการประมวลผลตามคำสั่งที่มีอยู่ที่เครื่องเสียก่อน แล้วจึงค่อยส่งผลลัพธ์ที่ได้ให้เรา ผลลัพธ์ที่ได้นั้นก็คือเว็บเพจที่เราเห็นนั่นเอง ถือได้ว่า PHP เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้เราสามารถสร้าง Dynamic Web pages (เว็บเพจที่มีการโต้ตอบกับผู้ใช้) ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น (<http://www.mindphp.com/คู่มือ/73-คืออะไร/2127-php-คืออะไร.html>)

### 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 ในติดต่อกับฐานข้อมูลและแสดงผลแบบ ดอนามิกและอื่นๆ มากกว่า 50,000 เว็บไซต์

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

PHP3 ออกมาในช่วงระหว่างเดือน มิถุนายน 1997 ถึง 1999 ได้ออกสู่สายตาของนักโปรแกรมเมอร์ มีคุณสมบัติเด่นคือสนับสนุนระบบปฏิบัติการทั้ง Window 95/98/ME/NT, Linux และเว็บเซิร์ฟเวอร์ 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> ) ในเวอร์ชันนี้จะเป็น compile script ซึ่งในเวอร์ชันหน้าจะเป็น embed script interpreter ในปัจจุบันมีคนได้ใช้ PHP สูงกว่า 5,100,000 เว็บไซต์ 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](http://www.php.net)

PHP เป็นเรื่องง่ายที่จะเรียนรู้และทำงานได้อย่างมีประสิทธิภาพด้านเซิร์ฟเวอร์

## การติดตั้ง PHP5

ก่อนการใช้งาน PHP สามารถตรวจสอบเครื่องคอมพิวเตอร์ดังนี้

1. ค้นหาเว็บไซต์ที่มีการสนับสนุน PHP และ MySQL
2. ติดตั้งเว็บเซิร์ฟเวอร์บนพีซีจากนั้นติดตั้ง PHP และ MySQL

.....เพิ่มเติม

## รูปแบบของ PHP

- PHP จะทำงานบนเซิร์ฟเวอร์ (Server) และถูกส่งกลับไปยังเบราว์เซอร์ (Browser) ในรูปแบบโค้ด HTML (Hypertext Markup Language)

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

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

- เครื่องหมายอ้อมภาค (;) คือ เครื่องหมายที่ใช้สิ้นสุดคำสั่งในแต่ละบรรทัดแสดงดังตัวอย่าง

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

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

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

## ตัวแปร (Variable)

ตัวแปร คือ สิ่งที่ใช้เก็บค่าของข้อมูลในหน่วยความจำ เพื่อนำไปประมวลผล ซึ่งตัวแปรจะขึ้นต้นด้วยเครื่องหมายดอลลาร์ \$ 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;
- ไม่ตั้งชื่อซ้ำคำสั่ง 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	virtual	use
class	function	static	xor	clone (5)
continue	global	switch	while	old fuction (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](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 ฟังก์ชันสามารถกล่าวได้ดังนี้

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 ทำงานเหมือนฟังก์ชัน ทำให้ print ใช้ในรูปประโยคที่ซับซ้อนได้ ส่วน echo ไม่สามารถทำได้ เช่น

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

### ตัวอย่างการใช้ echo

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

### ตัวอย่างการใช้ print

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

## ชนิดของข้อมูล (Data Type)

ตัวแปรสามารถจัดเก็บข้อมูลประเภทต่างๆ และชนิดข้อมูลที่แตกต่างกันสามารถทำสิ่งต่างๆ แสดงตัวตารางดังต่อไปนี้

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
    $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 (ตามที 8 - นำหน้าด้วย 0)

ตัวอย่างดังนี้

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

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

**Boolean** คือ ข้อมูลที่เก็บค่าความเป็นจริง คือ TRUE กับค่าความเป็นเท็จ คือ FALSE เช่น

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

**Array** คือ ตัวแปรชุดเก็บค่าของข้อมูลได้หลายค่า เช่น

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

**Object** คือ ตัวแปรประเภทหนึ่งทีสร้างมาจากคลาสหรือ *Class Instance* ออบเจ็กต์มีการอ้างอิงถึงสมาชิกของมันที่เป็นตัวแปรและเมธอด

ตัวอย่าง

```
<?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

โค้ด	ผลลัพธ์
<pre>&lt;?php \$name = "Chalawan Wantong"; \$age = 5; \$GPA = 2.43; echo "\$name &lt;br&gt;"; echo "\$age &lt;br&gt;"; echo "\$GPA "; ?&gt;</pre>	<p>Chalawan Wantong 5 2.43</p>

### 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 <code>\$x = 5; \$y = 2;</code>	ผลลัพธ์
+	บวก	<code>echo \$x + \$y;</code>	7
-	ลบ	<code>echo \$x - \$y;</code>	3
*	คูณ	<code>echo \$x * \$y;</code>	10
/	หาร	<code>echo \$x / \$y;</code>	2.5
%	หารเอาค่าเศษ	<code>echo \$x % \$y;</code>	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
$x = y$	$x = y$	<code>echo \$x = \$y;</code>	2
$x += y$	$x = x + y$	<code>echo \$x += \$y;</code>	7
$x -= y$	$x = x - y$	<code>echo \$x -= \$y;</code>	3
$x *= y$	$x = x * y$	<code>echo \$x *= \$y;</code>	10
$x /= y$	$x = x / y$	<code>echo \$x /= \$y;</code>	2.5
$x \% = y$	$x = x \% y$	<code>echo \$x %= \$y;</code>	1

### ตัวดำเนินการเชิงเปรียบเทียบ (Comparison operators)

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

ตัวดำเนินการ	ความหมาย	ตัวอย่าง กำหนดให้ $x = 5; y = 2;$ $z = '5';$	ผลลัพธ์
$==$	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าเท่ากัน	<code>echo \$x == \$z;</code>	true
$===$	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าเท่ากัน <b>และ</b> ประเภทตัวแปรเหมือนกัน	<code>echo \$x === \$z;</code>	false
$!=$	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน	<code>echo \$x != \$z;</code>	false
$<>$	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน	<code>echo \$x &lt;&gt; \$y;</code>	true
$!==$	จะเป็นจริงก็ต่อเมื่อตัวแปรมีค่าไม่เท่ากัน หรือ ประเภทตัวแปรไม่เหมือนกัน	<code>echo \$x !== \$z;</code>	true
$>$	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่ามากกว่า	<code>echo \$x &gt; \$y;</code>	true
$<$	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่าน้อยกว่า	<code>echo \$x &lt; \$y;</code>	false
$>=$	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่ามากกว่า หรือเท่ากับ	<code>echo \$x &gt;= \$y;</code>	true
$<=$	จะเป็นจริงก็ต่อเมื่อตัวแปรข้างซ้ายมีค่าน้อยกว่า หรือเท่ากับ	<code>echo \$x &lt;= \$y;</code>	false

### ตัวดำเนินการเพิ่มค่าหรือลดค่า (Increment/Decrement operators)

ตัวดำเนินการเพิ่มค่าหรือลดค่า ใช้เพิ่มค่าหรือลดค่าตัวแปรประเภทตัวเลขทีละ 1 ค่า

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

### ตัวดำเนินการเชิงตรรกะ (Logical operators)

ตัวดำเนินการเชิงตรรกะ ใช้กับคำสั่งที่เป็นเงื่อนไขเพื่อตรวจสอบว่าเป็นจริงหรือเป็นเท็จ

ตัวดำเนินการ	ชื่อ	ตัวอย่าง กำหนดให้ <code>\$x = true;</code> <code>\$y = false;</code>	ความหมาย	ผลลัพธ์
<code>and</code>	และ	<code>\$x and \$y;</code>	จะเป็นจริงก็ต่อเมื่อ <code>\$x, \$y</code> เป็นจริงทั้งสอง	<code>false</code>
<code>or</code>	หรือ	<code>\$x or \$y;</code>	จะเป็นจริงก็ต่อเมื่อ <code>\$x, \$y</code> เป็นจริงตัวใดตัวหนึ่ง	<code>true</code>
<code>xor</code>	xor	<code>\$x xor \$y;</code>	จะเป็นจริงก็ต่อเมื่อ <code>\$x, \$y</code> เป็นจริงตัวใดตัวหนึ่ง และห้ามเป็นจริงทั้ง 2 ตัว	<code>true</code>
<code>&amp;&amp;</code>	และ	<code>\$x &amp;&amp; \$y;</code>	จะเป็นจริงก็ต่อเมื่อ <code>\$x, \$y</code> เป็นจริงทั้งสอง	<code>false</code>
<code>  </code>	หรือ	<code>\$x    \$y;</code>	จะเป็นจริงก็ต่อเมื่อ <code>\$x, \$y</code> เป็นจริงตัวใดตัวหนึ่ง	<code>true</code>
<code>!</code>	ไม่ใช่	<code>!\$x;</code>	จะเป็นจริงก็ต่อเมื่อ <code>\$x</code> เป็นเท็จ	<code>false</code>

### ตัวดำเนินการเชิงข้อความ (String operators)

ตัวดำเนินการเชิงข้อความ ใช้งานโดยการเอาข้อความสองข้อความเรียงต่อกัน มี 2 รูปแบบ ดังนี้

เครื่องหมายจุด (.) คือ การนำข้อความทั้ง 2 ข้อความมาเรียงต่อกันตัวอย่างเช่น

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

```
?>
```

### ตัวดำเนินการเชิงอาร์เรย์ (Array operators)

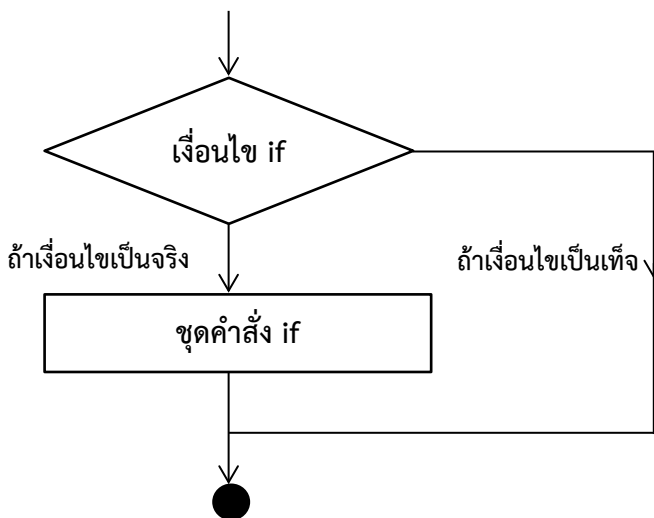
ใช้โดยการนำตัวแปรอาร์เรย์ 2 ตัวมาเปรียบเทียบกับกันโดยใช้เครื่องหมายดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
	กำหนดให้	<pre>\$x = array("a" =&gt; "red", "b" =&gt; "green"); \$y = array("c" =&gt; "blue", "d" =&gt; "yellow");</pre>	
+	นำค่าของ 2 ตัวแปรอาร์เรย์รวมกัน	<pre>print_r(\$x + \$y);</pre>	ค่าทั้งหมดของตัวแปร x และ y
==	จะเป็นจริงก็ต่อเมื่อตัวแปร มีค่าเท่ากัน	<pre>var_dump(\$x == \$y);</pre>	<b>false</b>
===	จะเป็นจริงก็ต่อเมื่อตัวแปร มีค่าเท่ากัน <b>และ</b> ประเภทตัวแปรเหมือนกัน	<pre>var_dump(\$x === \$y);</pre>	<b>false</b>
!=	จะเป็นจริงก็ต่อเมื่อตัวแปร มีค่าไม่เท่ากัน	<pre>var_dump(\$x != \$y);</pre>	<b>true</b>
<>	จะเป็นจริงก็ต่อเมื่อตัวแปร มีค่าไม่เท่ากัน	<pre>var_dump(\$x &lt;&gt; \$y);</pre>	<b>true</b>
!==	จะเป็นจริงก็ต่อเมื่อตัวแปร มีค่าไม่เท่ากัน หรือ ประเภทตัวแปรไม่เหมือนกัน	<pre>var_dump(\$x !== \$y);</pre>	<b>true</b>

## เงื่อนไข if else elseif switch

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

1. **เงื่อนไขทางเลือกเดียว (if)** เป็นการกำหนดเงื่อนไขทิศทางเดียว โดยที่โปรแกรมจะทำงานตามเงื่อนไขในกรณีที่เงื่อนไขเป็นจริงเท่านั้นซึ่งมีรูปแบบดังนี้



ภาพที่ 2 เงื่อนไขแบบทางเลือกเดียว

### รูปแบบคำสั่ง

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

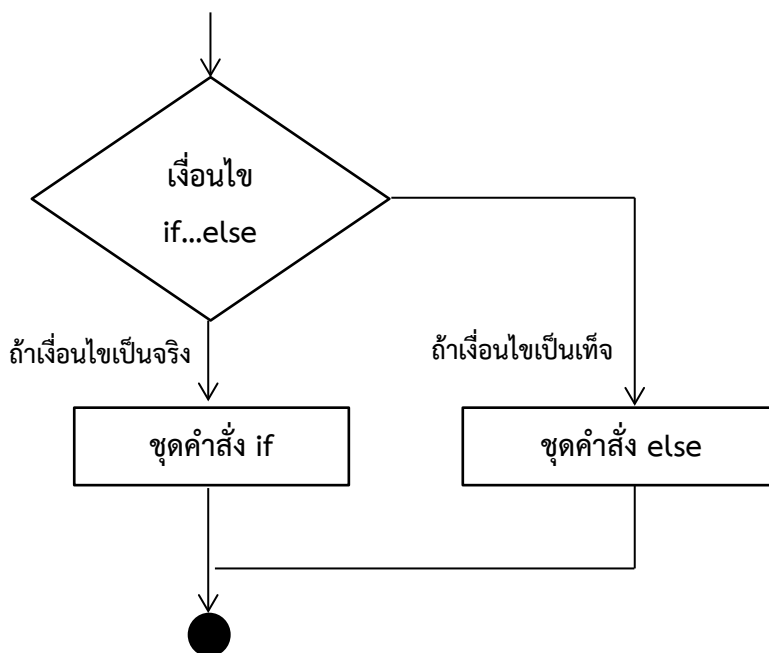
### ตัวอย่าง

```
<?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
    $score = 40;
    if($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
    $score = 70;
    if($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
    $favcolor = "red";
    switch ($favcolor) {
        case "red":
            echo "สีที่คุณชอบ คือ สีแดง" ;
            break;
        case "blue":
            echo "สีที่คุณชอบ คือ สีฟ้า" ;
            break;
        case "green":
            echo "สีที่คุณชอบ คือ สีเขียว" ;
            break;
        default:
            echo "สีที่คุณชอบนอกจากสีแดง สีฟ้า และ สีเขียว" ;
    }
?>
```

ผลลัพธ์

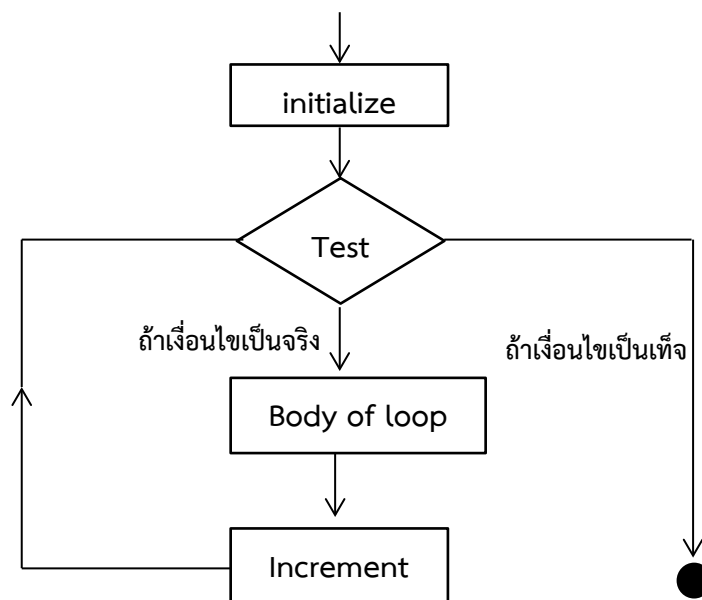
สีที่คุณชอบ คือ สีแดง

### 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

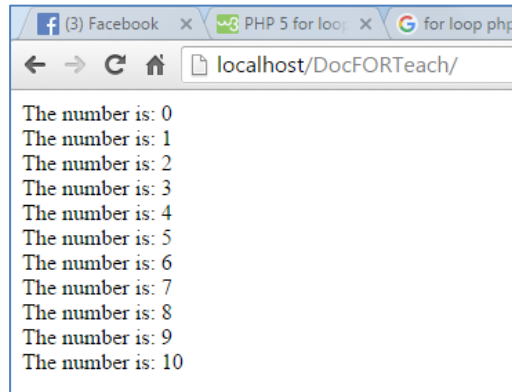
รูปแบบคำสั่ง

```
for ( init counter; test counter; increment counter) {
    code to be executed;
}
```

ตัวอย่างที่ 1 จงแสดงเลข 0-10

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

### ผลลัพธ์ 1



ภาพที่ 5 ผลลัพธ์การทำงานของลูป For

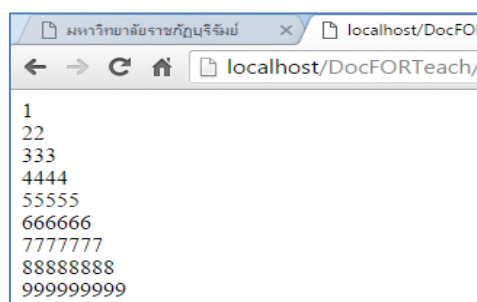
### ตัวอย่างที่ 2 จงแสดงจำนวนเต็มที่เป็นเลขคี่ตั้งแต่เลข 1-10

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

### ผลลัพธ์ที่ 2

13579

### ตัวอย่างที่ 3 จงเขียนโค้ดให้แสดงผลดังต่อไปนี้



ภาพที่ 6 ผลลัพธ์ของตัวอย่างที่ 3

### คำตอบตัวอย่างที่ 3

```
<?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

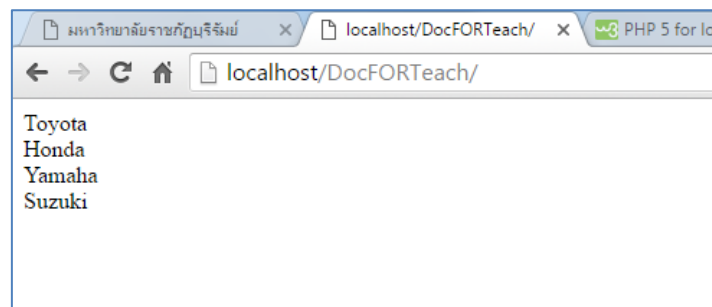
รูปแบบ

```
foreach ($array as $value) {
    code to be executed;
}
```

ตัวอย่าง

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

ผลลัพธ์



ภาพที่ 7 ผลลัพธ์ของ foreach loop