

บทที่ 5

PHP เบื้องต้น

A person wearing a white button-down shirt is shown from the chest up, with their arms slightly outstretched. The background is a composite image of a dense city skyline, featuring prominent skyscrapers like the Empire State Building. The image is split diagonally, with the cityscape on the left and a solid blue background on the right.

Agenda

1. PHP Syntax
2. PHP Operation
3. PHP Conditional Statements
4. PHP Looping
5. PHP Array
6. PHP String
7. PHP Function
8. การเขียน PHP ร่วมกับ HTML การส่งข้อมูลด้วย Get และ Post



PHP Syntax

- ▶ PHP คือ ภาษาโปรแกรม (Programming Language) ที่ทำงานฝั่ง Server (Server Side Script)
- ▶ ถูกประมวลผลที่ฝั่งเซิร์ฟเวอร์ ผู้ใช้งานจะมองไม่เห็น Code ในเพจ เพราะ code จะถูกประมวลผลจนหมดในฝั่งเซิร์ฟเวอร์
- ▶ เป็น open source
- ▶ โดยสามารถดาวน์โหลดได้จาก
- ▶ ปัจจุบันอยู่ที่ Version 8.1 (25 Nov 2021)

เริ่มต้นเขียน PHP

- Web Server คือ **Apache**

- Web Server package

 - Appserv

 - Xampp

 - ฯลฯ

- การรันโปรแกรม รันผ่าน Server โดยพิมพ์

```
http://localhost/ชื่อโฟลเดอร์/ชื่อไฟล์.php
```

```
http://127.0.0.1/ชื่อโฟลเดอร์/ชื่อไฟล์.php
```


เริ่มต้นเขียน PHP

- Code php ต้องเขียนอยู่ภายใต้คำสั่ง `<?php` และจบด้วย `?>`
- ไฟล์ของ code php ต้อง**นามสกุล .php**
- ปิดท้ายคำสั่ง php ด้วย `;`
- Comment ใน php ใช้ `//` หรือ `/*...*/`
- **Keyword class function** ใน php สามารถใช้เป็นตัวพิมพ์เล็กหรือพิมพ์ใหญ่ก็ได้ (**NOT case-sensitive**) เช่น echo Echo ECHO
- **ตัวแปรเป็น Case Sensitive** เช่น \$str ไม่เท่ากับ \$Str

การเขียน PHP ร่วมกับ HTML

- แบบที่ 1 แทรก php ลงใน โค้ด html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
        <?php
            echo "hello world";
        ?>
    </body>
</html>
```

การเขียน PHP ร่วมกับ HTML

- แบบที่ 2 แทรก html ลงในโค้ด php

```
<?php
echo `<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
        hello world
    </body>
</html>`;
?>
```


PHP Variables

- การประกาศตัวแปรใน php ใช้เครื่องหมาย \$ นำหน้าชื่อตัวแปร
- ชื่อตัวแปรต้องขึ้นต้นด้วยตัวอักษรห้ามขึ้นต้นด้วยตัวเลข
- ชื่อตัวแปรเป็น **case-sensitive**

```
<?php
    $txt = "Hello world!";
    $x = 5;
    $y = 10.5;

?>
```

PHP echo and print Statements

- การใช้คำสั่ง echo และคำสั่ง print ในภาษา php เป็นคำสั่งที่ใช้ในการแสดงผลที่หน้าจอ
- สามารถใช้ร่วมกับภาษา html ได้
- คำสั่ง echo ต่างจาก print ตรงที่สามารถรับ parameter ได้มากกว่า 1 ตัว

```
<?php
    print "Hello World!"; //ใช้ได้
    echo "Hello World!"; //ใช้ได้
    print ("Hello World!"); //ใช้ได้
    echo ("Hello World!"); //ใช้ได้
    print "test","again";// ใช้ไม่ได้
    echo "test","again"; // ใช้ได้
    print("test","again"); //ใช้ไม่ได้
    echo ("test","again"); //ใช้ไม่ได้
?>
```

PHP Data Types

- String สามารถใช้ได้ทั้ง ‘ (Single quotes) และ “ (Double quotes)
- Integer
- Float
- Boolean
- Array

```
<?php  
    var_dump($a); //ตรวจสอบ type  
?>
```

```
<?php  
    $a = "Hello world!";  
    $b = 'Hello world!';  
    $c = 5985; //Integer  
    $d = 10.365; //Float  
    $e = true; //Boolean  
    $f = false; //Boolean  
    $g = array("Volvo", "BMW", "Toyota"); //Array  
?>
```



PHP Operation

1. Arithmetic operators
2. Assignment operators
3. Comparison operators
4. Increment/Decrement operators
5. Logical operators
6. String operators
7. Array Operators

1. Arithmetic operators are used with numeric values

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power (Introduced in PHP 5.6)

2. Assignment Operators are used with numeric values to write a value to a variable.

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus

3. Comparison Operators are used to compare two values (number or string)

Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type
!=	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y
<>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y

4. Increment / Decrement Operators

The PHP increment operators are used to increment a variable's value.
The PHP decrement operators are used to decrement a variable's value.

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x++</code>	Post-increment	Returns <code>\$x</code> , then increments <code>\$x</code> by one
<code>--\$x</code>	Pre-decrement	Decrements <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x--</code>	Post-decrement	Returns <code>\$x</code> , then decrements <code>\$x</code> by one

5. Logical Operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	<code>\$x and \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
or	Or	<code>\$x or \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
xor	Xor	<code>\$x xor \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true, but not both
<code>&&</code>	And	<code>\$x && \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
<code> </code>	Or	<code>\$x \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
<code>!</code>	Not	<code>!\$x</code>	True if <code>\$x</code> is not true

ความแตกต่างระหว่างฟังก์ชัน OR และ XOR

<u>Conditon1</u>	<u>Condition2</u>	OR	XOR
TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	TRUE	TRUE
FALSE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE

6. String Operators that are specially designed for strings.

Operator	Name	Example	Result
.	Concatenation	<code>\$txt1 . \$txt2</code>	Concatenation of <code>\$txt1</code> and <code>\$txt2</code>
<code>.=</code>	Concatenation assignment	<code>\$txt1 .= \$txt2</code>	Appends <code>\$txt2</code> to <code>\$txt1</code>

7. Array Operators are used to compare arrays.

7. Array Operators are used to compare arrays.

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

<?php
$x = array("a" => "red", "b" => "green");
$y = array("c" => "blue", "d" => "yellow");

print_r($x + $y); // union of $x and $y
?>

</body>
</html>
```

ผลลัพธ์ Array ([0] => red [1] => green [a] => blue [b] => yellow)



PHP Conditional Statements

■ if statement

```
If (เงื่อนไข) {  
    คำสั่งเมื่อเงื่อนไขเป็นจริง  
}
```

```
<?php  
    $t = 18;  
  
    if ($t < 20) {  
        echo "Have a good day!";  
    }  
  
?>
```

04

PHP Conditional Statements

■ if...elseif...else statement

```
if (เงื่อนไข1) {  
    คำสั่งเมื่อเงื่อนไขเป็นจริง  
}else if (เงื่อนไข2) {  
    คำสั่งเมื่อเงื่อนไข2เป็นจริง  
}else{  
    คำสั่งเมื่อเงื่อนไข1 และ 2 เป็นเท็จ  
}
```

```
<?php  
  
    $t = 18;  
  
    if ($t < 20) {  
        echo "t น้อยกว่า 20";  
    }else if ($t == 20) {  
        echo "t เท่ากับ 20";  
    }else {  
        echo "t มากกว่า 20";  
    }  
  
?>
```

■ switch statement

```
switch (n) {  
    case label1:  
        ทำคำสั่งตรงนี้เมื่อ n=label1;  
        break;  
    case label2:  
        ทำคำสั่งตรงนี้คำสั่งเมื่อ n=label2;  
        break;  
    case label3:  
        ทำคำสั่งตรงนี้คำสั่งเมื่อ n=label3;  
        break;  
    ...  
    default:  
        ทำคำสั่งตรงนี้คำสั่งเมื่อ n ไม่เท่ากับ label1 label2 label3;  
}  

```

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "สีแดง";
        break;
    case "blue":
        echo "สีน้ำเงิน";
        break;
    case "green":
        echo "สีเขียว";
        break;
    default:
        echo "ไม่ใช่สีแดง สีน้ำเงิน สีเขียว";
}
?>
```

ตัวอย่าง switch statement

เงื่อนไข	ค่า
คะแนน ≥ 80	A
คะแนน ≥ 60	C
คะแนน ≥ 50	D
นอกนั้น	F



PHP Looping

- **while - loops** คำสั่งที่อยู่ใน { } ปีกกาจะทำงานเมื่อเงื่อนไขเป็นจริง

```
<?php
$x = 1;

while ($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
```

- **do...while Loop** จะทำงานรอบแรกก่อนเสมอ แล้วจึงตรวจสอบเงื่อนไข

```
<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```

■ For loop ใช้เมื่อรู้จำนวนรอบที่แน่นอน

```
for (init counter; test counter; in/decrement counter)  
{  
    code to be executed;  
}
```

■ **init counter** ค่าเริ่มต้น

■ **test counter** เงื่อนไข

■ **in/decrement counter** การเพิ่ม/ลดค่าในแต่ละรอบ

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



PHP *Array*

05

PHP Array

```
$cars1 = "Volvo";  
$cars2 = "BMW";  
$cars3 = "Toyota";
```



```
$cars = array("Volvo", "BMW", "Toyota");  
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

■ Array ใน PHP มี 2 ประเภทคือ

- **Indexed arrays** – คือ array ที่ใช้ตัวเลขเป็น index

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

- **Associative arrays** – คือ array ที่ใช้ชื่อเป็นคีย์

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

- Indexed arrays – คือ array ที่ใช้ตัวเลขเป็น index

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

echo "I like " . $cars[0] . ", " .
$cars[1] . " and " . $cars[2] . ".";
?>
```

■ Associative arrays – คือ array ที่ใช้ชื่อเป็นคีย์

```
<?php
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
echo "Peter is " . $age['Peter'] . " years old.";
```

หรือเขียนในชุดเดียวกันดังนี้

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
```


- **foreach Loop** ใช้สำหรับการแสดงค่าใน Array เท่านั้น Indexed arrays

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

- ตัวอย่าง

```
<?php  
$animal = array("dog", "cat", "fish", "pig");  
  
foreach ($animal as $value) {  
    echo "$value <br>";  
}  
?>
```

Associative arrays

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37",
"Joe"=>"43");

foreach($age as $x => $val) {
    echo "$x = $val<br>";
}

?>
```

■ Multidimensional arrays – คือ array หลายมิติ

```
$numbers = array(  
    [10, 50],  
    [43, 7, 5],  
    [10],  
    [20, 20],  
    [80, 30, 100, 120]  
);
```

	Column 1	Column 2	Column 3	Column 4
Row 1	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 2	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 3	a[2][0]	a[2][1]	a[2][2]	a[2][3]

```
for ($i = 0; $i < count($numbers); $i++) {  
    $sum = 0;  
    echo "Sum of ";  
    for ($j = 0; $j < count($numbers[$i]); $j++) {  
        $sum += $numbers[$i][$j];  
        echo $numbers[$i][$j] . " ";  
    }  
    echo "= $sum\n";  
}
```

สร้างตาราง 7 คอลัมน์ 6 แถว โดยใช้ for loop

```

echo '<table border="1">
<thead><tr>
    <th>อา</th>
    <th>จ</th>
    <th>อ</th>
    <th>พ</th>
    <th>พฤ</th>
    <th>ศ</th>
    <th>ส</th>
</tr></thead>
<tbody>';
$d=1;
for [REDACTED]
{
echo '<tr>';
    for [REDACTED]
    {
        echo '<td>' . $d . '</td>';
        $d++;
    }
    echo '</tr>';
}
echo '</tbody> </table>';

```

อา	จ	อ	พ	พฤ	ศ	ส
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35



PHP String

- **strlen** ฟังก์ชันการนับจำนวนตัวอักษร

```
echo strlen("Hello world!"); // outputs 12
```

- **str_word_count** ฟังก์ชันการนับจำนวนคำ

```
echo str_word_count("Hello world!"); // outputs 2
```

- **strrev** ฟังก์ชันการ reverse string

```
echo strrev("Hello world!"); // outputs !dlrow olleH
```

- **strpos** ฟังก์ชันการหาตำแหน่งเริ่มต้นของคำ

```
echo strpos("Hello world!", "world"); // outputs 6
```

- **str_replace** ฟังก์ชันการแทนที่คำ

```
echo str_replace("world", "Dolly", "Hello world!");  
// outputs Hello Dolly!
```



PHP Function

```
function functionName ($Argument, ...) {  
    code to be executed;  
    return value;  
}
```

- **functionName** คือ ชื่อฟังก์ชันที่สร้างขึ้น
- **Argument** คือ ตัวแปรที่รับเข้าไปในฟังก์ชัน สามารถมีได้มากกว่า 1 ตัว โดยใช้ คอมม่า (,) คั่น
- **Return** ส่งค่ากลับไปนอกฟังก์ชัน จะมีหรือไม่มีก็ได้ หากมีการ Return ค่าเวลาเรียกใช้ฟังก์ชัน ต้องมีตัวแปรมารับ
- การเรียกใช้ฟังก์ชัน ทำได้โดยเรียกชื่อฟังก์ชัน เช่น
functionName();

■ การสร้างฟังก์ชันใน PHP

■ ฟังก์ชันที่ไม่มีการรับพารามิเตอร์

```
function functionName() {  
    code to be executed;  
}
```

■ ฟังก์ชันที่มีการรับพารามิเตอร์

```
function functionName($Argument) {  
    code to be executed;  
}
```

■ ฟังก์ชันที่มีการส่งค่ากลับ (return)

```
function functionName($Argument) {  
    code to be executed;  
    return $parameter; //ส่งค่ากลับไปยังบรรทัดที่เรียกใช้ฟังก์ชัน  
}
```

■ ตัวอย่างการใช้งาน Function (ไม่มีการรับส่งค่าคืน)

```
<?php  
function writeMsg() {  
    echo "Hello world!";  
}  
  
writeMsg(); // call the function  
?>
```

- ตัวอย่างการใช้งาน Function (รับค่า 1 ตัวแปร และแสดงผลเลย)

```
<?php
function familyName ($fname) {
    echo "$fname Refsnes.<br>";
}

familyName ("Jani"); // call the function
familyName ("Hege"); // call the function
?>
```

■ ตัวอย่างการใช้งาน Function (รับค่า 2 ตัวแปร และแสดงค่าเลย)

```
<?php
function familyName($fname, $year) {
    echo "$fname Refsnes. Born in $year
<br>";
}

familyName ("Hege", "1975");
familyName ("Stale", "1978");
?>
```

- ตัวอย่างการใช้งาน Function (กำหนดค่าปริยายหากไม่มีค่าส่งมา)

```
<?php
function setHeight($minheight = 50) {
    echo "The height is : $minheight <br>";
}

setHeight(350);
setHeight(); // will use the default value of 50
?>
```

ตัวอย่างการใช้งาน Function (ที่มีการรับค่าและส่งค่ากลับ)

```
<?php
function sum($x, $y) {
    $z = $x + $y;
    return $z;
}

echo "5 + 10 = " . sum(5,10) . "<br>";
echo "7 + 13 = " . sum(7,13) . "<br>";
echo "2 + 4 = " . sum(2,4);
?>
```

ผลลัพธ์

```
5 + 10 = 15
7 + 13 = 20
2 + 4 = 6
```

- ตัวอย่างการใช้งาน Function (ที่มีการรับค่า 2 ค่า และส่งค่ากลับ 2 ค่า)

```
<?php
function cal($x, $y) {
    $ans1 = $x + $y;
    $ans2 = $x - $y;
    return array($ans1, $ans2);
}

$ans = cal(5,10) . "<br>";
echo "ผลลัพธ์คือ" . $ans[0]. "และ" . $ans[1];
?>
```

ผลลัพธ์ คือ 15 และ -5

08

การเขียน PHP ร่วมกับ HTML

การส่งข้อมูลด้วย Get และ Post

08

การเขียน PHP ร่วมกับ HTML การส่งข้อมูลด้วย Get และ Post

- Form การกรอกข้อมูล HTML/PHP

M-Tourism

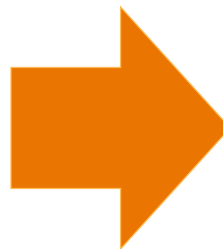
เพิ่มข้อมูล

ชื่อ
ใส่ชื่อ... name = "fname"

นามสกุล
ใส่นามสกุล...

อายุ
ใส่อายุ...

บันทึกข้อมูล



```
<html>
<body>

Welcome <?php echo
    $_POST["fname"] ;?>

</body>
</html>
```



08

การเขียน PHP ร่วมกับ HTML การส่งข้อมูลด้วย Get และ Post

- **\$_GET** เป็นตัวแปร ใช้สำหรับรับค่าผ่าน method GET หรือรับค่าผ่าน URL

```
<form action="showget.php" method="get">  
    <input type="text" name="fname">  
    <input type="submit">  
</form>
```



```
<?php  
    echo $_GET['fname'];  
?>
```

ผลลัพธ์ : ค่าที่ป้อนจาก Textbox ชื่อ fname ส่งผ่าน URL มา

- **\$_POST** เป็นตัวแปร ใช้สำหรับรับค่าผ่าน method POST หรือ รับค่าผ่าน Form HTML นั้นเอง

```
<form action="show2.php" method="post">  
  <input type="text" name="fname">  
  <input type="text" name="lname">  
  <input type="submit">  
</form>
```



```
<?php  
    echo $_POST['fname'];  
    echo $_POST['lname'];  
?>
```

ผลลัพธ์ : ค่าที่ป้อนจาก Textbox ชื่อ fname และ lname

- Isset ตรวจสอบว่า มีตัวแปรนี้อยู่หรือไม่

```
if (isset ( $_POST [ 'name' ] ) )  
{  
    คำสั่งเมื่อเงื่อนไขเป็นจริง  
}
```

08

การเขียน PHP ร่วมกับ HTML การส่งข้อมูลด้วย Get และ Post

- หากต้องการส่งค่าทั้ง 2 แบบพร้อมกัน ให้ใช้ method POST เป็นตัวแปร ใช้สำหรับรับค่าผ่านฟอร์ม และใช้ method GET ส่งค่าตัวแปรใน Attribute action ผ่าน URL

```
<form action="test.php?a=123" method="post">  
    <input type="text" name="a">  
    <input type="submit">  
</form>
```



```
<?php  
    echo $_GET['a'];  
    echo $_POST['a'];  
?>
```

ผลลัพธ์ : ค่าที่ได้จาก ตัวแปร a ที่ส่งผ่าน URL และค่าตัวแปร a ที่ได้จาก Textbox ใน form



THANK YOU

ใบงานที่ 5

