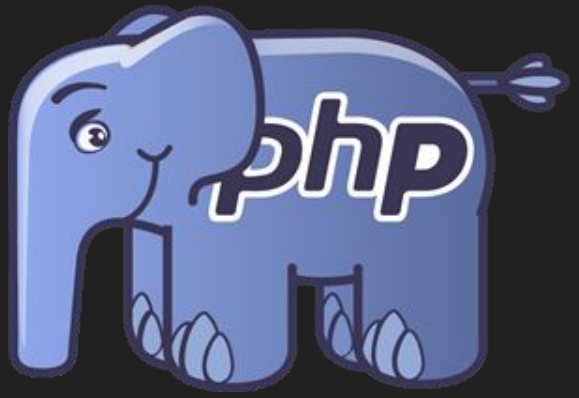


PHP เบื้องต้น

ฉบับปรับปรุง 2021

ติดตามผู้เขียน
ผ่านช่องทางยูทูป

https://www.youtube.com/channel/UCQ1r_4x-P-fETLIU4pqf98w

https://www.facebook.com/KongRuksiamTutorial/

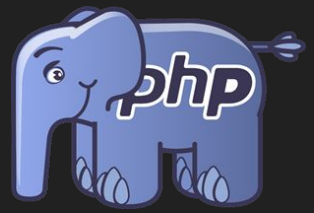
หรือสแกน QRCODE

เข้าเรียนเนื้อหา PHP ได้ที่

http://bit.ly/2XEOUXk

HTML ร่วมกับ PHP

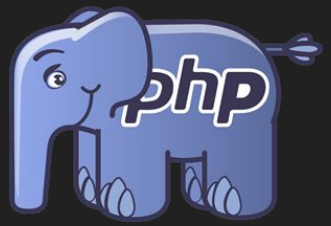
```
<html>
<head>
        <title>PHP เบื้องต้น</title>
</head>
<body>
        <p>html</p>
<?php
                echo “<h2>แสดงผลทางจอภาพ</h2>”;
        phpinfo( ); // แสดงรายละเอียดของ PHP
?>
</body>
</html>
```



print() และ echo()

เป็น function ที่ใช้ในการแสดงข้อมูลผ่าน Browser ยกตัวอย่าง เช่น

- echo “kongruksiam”;
- print “kongruksiam”;



print() และ echo() ต่างกันอย่างไร

1. print มีการส่งกลับมา (return) แต่ echo จะไม่มีการส่งค่ากลับมา (void)
2. print สามารถระบุค่าได้ตัวเดียว แต่ echo มีได้หลายตัว
3. echo เร็วกว่า print

จะลงรายละเอียดในหัวข้อ function

ตัวแปร (Variables)

ใช้สำหรับเก็บข้อมูลในหน่วยความจำเพื่อใช้ประมวลผล
เช่น ตัวเลข ตัวอักษร หรือชุดข้อความ หรือกลุ่มข้อมูล (Array)

การนิยามตัวแปร

โครงสร้าง

$ ชื่อตัวแปร = ค่าของตัวแปร

```
$name = “kongruksiam”
$age= 20;
$year = $age;
```

ให้นำค่าทางขวามือของเครื่องหมาย = ไปเก็บไวในตัวแปรที่อยู่ด้านซ้ายมือ

กฎการตั้งชื่อตัวแปร

1. สามารถเป็นได้ทั้งตัวอักษร ตัวเลข สัญลักษณ์ _(underscores) และ $(dollar sign)
2. อักษรตัวแรกห้ามเป็นตัวเลข
3. Case Sensitive ตัวพิมพ์เล็ก-พิมพ์ใหญ่ มีความหมายต่างกัน
4. ไม่ซ้ำกับ built-in function (ฟังก์ชั่นมาตรฐานใน PHP)
5. ตัวแปรขึ้นต้นด้วย $(dollar sign)
6. ตัวแปรไม่สามารถเว้นว่าง หรือเคาะเว้นวรรคได้

แสดงผลตัวแปร

```
$name = “kongruksiam”;
$age= 20-5;
$year = $age;

echo “ชื่อ = ”.$name.”<br>”;
echo “อายุ = ”.$age.”<br>”;
```

ชนิดข้อมูลของตัวแปร

1. **Integer** คือเลขจำนวนเต็ม เช่น 1, 2, 3...(ฐาน 10 , 8 , 16) เป็นต้น
2. **Float/Double** คือเลขจำนวนจริง เช่น 1.23, 3.14,... เป็นต้น
3. **Boolean** คือข้อมูลทางตรรกศาสตร์มีค่า จริง (True) , เท็จ (False)
4. **String** คือตัวอักษรหรือชุดข้อความเขียนภายในเครื่องหมาย “”
5. **Array** คือชุดหรือกลุ่มของข้อมูลที่มีชนิดข้อมูลเดียวกัน
6. **Object** คือการกำหนดให้ตัวแปรนั้นเก็บคุณสมบัติของ Object (Attribute & Method) โดยการประกาศใช้งานผ่าน Class (OOP)

แสดงชนิดข้อมูลของตัวแปร

gettype() , settype()

เช่น $total=3.14;

```
echo gettype($total);
settype($total,”integer’);
echo gettype($total);
```

Type Casting

Casting คือ การเปลี่ยนชนิดของข้อมูลให้เป็นชนิดที่ต้องการ โดยใส่ชนิดข้อมูลของตัวแปรไว้ในวงเล็บหน้าตัวแปรที่ต้องการจะเปลี่ยนชนิดของข้อมูล

```
$a=10.5;
$b=20.3;
$c=$a+$b;
echo $c;
```

```
$a=(integer)$a;
$b=(integer)$b;
$c=$a+$b;
echo $c;
```

ค่าคงที่ (Constant)

การสร้างตัวแปรที่เก็บข้อมูลให้ไม่สามารถเปลี่ยนแปลงค่าได้

define(ชื่อตัวแปร , ค่าที่กำหนด)

define(title,”kongruksiam”);

define(pi,3.14);

ฟังก์ชั่นที่ทำงานเกี่ยวกับตัวแปร

- **isset** คือฟังก์ชั่นสำหรับตรวจสอบว่าตัวแปรมีการกำหนดค่าหรือไม่

ถ้ากำหนดจะมีค่าเป็น True (1) ถ้าไม่กำหนดจะมีค่าเป็น False

- **unset** ยกเลิกตัวแปรและคืนค่าให้หน่วยความจำ
- **empty** ฟังก์ชั่นสำหรับตรวจสอบว่าตัวแปรมีค่าว่างหรือเลขศูนย์หรือไม่

ถ้าเป็นค่าว่างจะเป็น True (1)ถ้าไม่เป็นค่าว่างจะเป็น False

- **is_null** ฟังก์ชั่นสำหรับตรวจสอบว่าตัวแปรมีค่าว่างหรือไม่
- **print_r()** ฟังก์ชั่นสำหรับแสดงค่าตัวแปร array
- **var_dump()** แสดงรายละเอียดตัวแปร

ตัวแปร Superglobal

เป็นตัวแปรที่รับค่าตัวแปรจาก browser กับ web server

- **$GLOBALS** เป็นการประกาศให้เป็นตัวแปร global เพื่อให้ทุกส่วนสามารถเรียกใช้งานได้เลย
- **$_SERVER** เก็บค่าต่างๆของ web server ที่กำลังทำงานอยู่
- **$_GET** เป็นตัวแปรแบบ Array ใช้เก็บค่าที่ส่งมากับ URL
- **$_POST** ใช้เก็บค่าที่ส่งมากับ Form แบบ post method
- **$ENV** ตัวแปรที่จัดเก็บสภาพแวดล้อมทั่วไปและค่าต่างๆของ SERVER
- **$_SESSION** เก็บตัวแปร session
- **$_COOKIE** เก็บตัวแปร cookie

ตัวดำเนินการ (Operator)

กลุ่มของเครื่องหมายหรือสัญลักษณ์ที่ใช้ในการเขียนโปรแกรม

A+B

1. ตัวดำเนินการ (Operator)
2. ตัวถูกดำเนินการ (Operand)

ตัวดำเนินการทางคณิตศาสตร์

Operator	คำอธิบาย
+	บวก
-	ลบ
*	คูณ
/	หาร
%	หารเอาเศษ

ตัวดำเนินการเปรียบเทียบ

Operator	คำอธิบาย
==	เท่ากับ
===	เหมือนกัน
!=	ไม่เท่ากับ
<>	ไม่เท่ากับ
!==	ไม่เหมือนกัน
>	มากกว่า
<	น้อยกว่า
>=	มากกว่าเท่ากับ
<=	น้อยกว่าเท่ากับ
<=>	spaceship (php7)

ตัวดำเนินการทางตรรกศาสตร์

Operator	คำอธิบาย
&&	AND
||	OR
!	NOT
and	เหมือนกับ && แต่ลำดับความสำคัญน้อยกว่า
or	เหมือนกับ || แต่ลำดับความสำคัญน้อยกว่า
not	เหมือนกับ ! แต่ลำดับความสำคัญน้อยกว่า

ตัวดำเนินการทางตรรกศาสตร์

a	!a	a	b	a && b	a || b
true	false	false	false	false	false
false	true	false	true	false	true
		true	false	false	true
		true	true	true	true

ตัวดำเนินการกับข้อความ (String)

1. . (concatenation) การต่อข้อความ
2. .= ต่อข้อความและกำหนดค่า

ตัวดำเนินการเพิ่มค่า - ลดค่า

Operator	รูปแบบการเขียน	ความหมาย
++ (Prefix)	++a	เพิ่มค่าให้ a ก่อน 1 ค่าแล้วนำไปใช้
++ (Postfix)	a++	นำค่าปัจจุบันใน a ไปใช้ก่อนแล้วค่อยเพิ่มค่า
-- (Prefix)	--b	ลดค่าให้ b ก่อน 1 ค่าแล้วนำไปใช้
-- (Postfix)	b--	นำค่าปัจจุบันใน b ไปใช้ก่อนแล้วค่อยลดค่า

Compound Assignment

Assignment	รูปแบบการเขียน	ความหมาย
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

ลำดับความสำคัญของตัวดำเนินการ

ลำดับที่	เครื่องหมาย	ลำดับการทำงาน
1	new	ไม่มีความสัมพันธ์
2	( )	ไม่มีความสัมพันธ์
3	[]	ไม่มีความสัมพันธ์
4	++ , --	ซ้ายไปขวา
5	* , / , %	ซ้ายไปขวา
6	+ , -	ซ้ายไปขวา
7	< , <= , > , >=	ซ้ายไปขวา
8	== , != , === <>	ไม่มีความสัมพันธ์
9	&& , ||	ซ้ายไปขวา
10	= , += , -= , *= , /= , %=	ขวาไปซ้าย

ตัวดำเนินการอื่นๆ

- comma (,) คั่น argument สำหรับรับส่งค่าในฟังก์ชั่น หรือสมาชิกใน array
- operator พิเศษ เช่น -> , new สำหรับการสร้าง class (oop)
- array operator
- ternary operator
- **Execution Operator** แทนด้วยสัญลักษณ์ `` (backticks) สำหรับใส่คำสั่ง command line server
- **Error Suppression Operator** (@) ส่วนที่ให้โปรแกรมมองข้ามกรณีที่มีข้อผิดพลาดเกิดขึ้น

กรณีศึกษา

1. 5+8 *9
2. 10 - 4+2
3. 10 - (2+1)
4. 5 * 2 – 40 / 5
5. 7+8/2+25

โครงสร้างควบคุม (Control Structure)

คือ กลุ่มคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรม

- แบบลำดับ (Sequence)
- แบบมีเงื่อนไข (Condition)
- แบบทำซ้ำ (Loop)

แบบมีเงื่อนไข (Condition)

กลุ่มคำสั่งที่ใช้ตัดสินใจในการเลือกเงื่อนไขต่างๆ ภายในโปรแกรมมาทำงาน

- if
- Switch..Case

รูปแบบคำสั่งแบบเงื่อนไขเดียว

- **if statement**

เป็นคำสั่งที่ใช้กำหนดเงื่อนไขในการตัดสินใจทำงานของโปรแกรม

ถ้าเงื่อนไขเป็นจริงจะทำตามคำสั่งต่างๆ ที่กำหนดภายใต้เงื่อนไขนั้นๆ

```
if(เงื่อนไข){
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;
}
```

รูปแบบคำสั่งแบบ 2 เงื่อนไข

```
if(เงื่อนไข){
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;
}else{
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ ;
}
```

ข้อควรระวังการเขียน if เพื่อตรวจสอบเงื่อนไข

```
if(เงื่อนไข){
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;
}if(เงื่อนไข){
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;
}
```

รูปแบบคำสั่งแบบหลายเงื่อนไข

```
if(เงื่อนไขที่ 1){
        คำสั่งเมื่อเงื่อนไขที่ 1 เป็นจริง ;
}elseif(เงื่อนไขที่ 2){
        คำสั่งเมื่อเงื่อนไขที่ 2 เป็นจริง ;
}elseif(เงื่อนไขที่ 3){
        คำสั่งเมื่อเงื่อนไขที่ 3 เป็นจริง ;
}else{
        คำสั่งเมื่อทุกเงื่อนไขเป็นเท็จ ;
}
```

if..else แบบลดรูป (Ternary Operator)

(เงื่อนไข) ? คำสั่งเมื่อเงื่อนไขเป็นจริง : คำสั่งเมื่อเงื่อนไขเป็นเท็จ;

```
if(เงื่อนไข){
    คำสั่งเมื่อเงื่อนไขเป็นจริง
}else{
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ
}
```

แบบมีเงื่อนไข (Condition)

กลุ่มคำสั่งที่ใช้ตัดสินใจในการเลือกเงื่อนไขต่างๆ ภายในโปรแกรมมาทำงาน

- **Switch..Case**

Switch เป็นคำสั่งที่ใช้กำหนดเงื่อนไขคล้ายๆกับ if แต่จะเลือกเพียงหนึ่งทางเลือกออกมาทำงานโดยนำค่าในตัวแปรมากำหนดเป็นทางเลือกผ่านคำสั่ง case

รูปแบบคำสั่ง

```
switch(สิ่งที่ต้องการตรวจสอบ) {

        case ค่าที่ 1 : คำสั่งที่ 1;
                break;
         case ค่าที่ 2 : คำสั่งที่ 2;
                break;
                .....
         case ค่าที่ N : คำสั่งที่ N;
                break;

        default : คำสั่งเมื่อไม่มีค่าที่ตรงกับที่ระบุใน case
}
```

***คำสั่ง break จะทำให้โปรแกรมกระโดดออกไปทำงานนอกคำสั่ง switch ถ้าไม่มีคำสั่ง break โปรแกรมจะทำคำสั่งต่อไปเรื่อยๆ จนจบการทำงาน

รูปแบบคำสั่ง

```
switch(month) {

        case 1 : echo “มกราคม”;
            break;
        case 2: echo “กุมภาพันธ์”;
        break;
                .....
         case ค่าที่ N : คำสั่งที่ N;
                 break;
        default : echo “ไม่พบเดือน”;
}
```

กำหนดให้ตัวแปร month เก็บตัวเลข

แบบทำซ้ำ (Loop)

กลุ่มคำสั่งที่ใช้ในการวนรอบ (loop) โปรแกรมจะทำงานไปเรื่อยๆจนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ จึงจะหยุดทำงาน

- While
- For
- Do..While

คำสั่งที่เกี่ยวข้องกับ Loop

- **break** ถ้าโปรแกรมพบคำสั่งนี้จะหลุดจากการทำงานในลูปทันที เพื่อไปทำคำสั่งอื่นที่อยู่นอกลูป

- **continue** คำสั่งนี้จะทำให้หยุดการทำงานแล้วย้อนกลับไปเริ่มต้นการทำงานที่ต้นลูปใหม่

- **exit** คำสั่งให้โปรแกรมหยุดทำงาน (ใช้ในกรณีที่มีข้อผิดพลาดเกิดขึ้นในโปรแกรม จะออกจากการทำงานของโปรแกรมทั้งหมด)

คำสั่ง While

- **While Loop**

จะทำงานตามคำสั่งภายใน while ไปเรื่อยๆเมื่อเงื่อนไขที่กำหนดเป็นจริง

```
while(เงื่อนไข){
    คำสั่งที่จะทำซ้ำเมื่อเงื่อนไขเป็นจริง ;
}
```

คำสั่ง For

- **For Loop**

เป็นรูปแบบที่ใช้ในการตรวจสอบเงื่อนไข มีการกำหนดค่าเริ่มต้น และเปลี่ยนค่าไปพร้อมๆกัน เมื่อเงื่อนไขในคำสั่ง for เป็นจริงก็จะทำงานตามคำสั่งที่แสดงไว้ภายในคำสั่ง for ไปเรื่อยๆ

โครงสร้างคำสั่ง

```
for(ค่าเริ่มต้นของตัวแปร; เงื่อนไข; เปลี่ยนแปลงค่าตัวแปร) {
        คำสั่งเมื่อเงื่อนไขเป็นจริง;
}
```

```
for($i = 1;$i<=10;$i++) {
    คำสั่งเมื่อเงื่อนไขเป็นจริง;
}
```

คำสั่ง Do..While

- **Do..While**

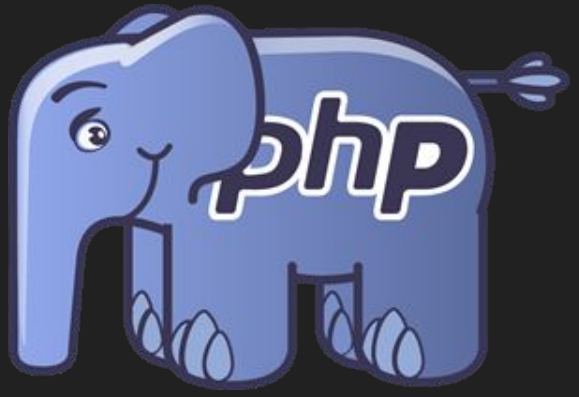
โปรแกรมจะทำงานตามคำสั่งอย่างน้อย 1 รอบ เมื่อทำงานเสร็จจะมาตรวจสอบเงื่อนไขที่คำสั่ง while ถ้าเงื่อนไขเป็นจริงจะวนกลับขึ้นไปทำงานที่คำสั่งใหม่อีกรอบ แต่ถ้าเป็นเท็จจะหลุดออกจากลูป

โครงสร้างคำสั่ง

```
do {
        คำสั่งต่างๆ เมื่อเงื่อนไขเป็นจริง;
 } while(เงื่อนไข);
```

ข้อแตกต่างและการใช้งาน Loop

- For ใช้ในกรณีรู้จำนวนรอบที่ชัดเจน
- While ใช้ในกรณีที่ไม่รู้จำนวนรอบ
- Do..while ใช้ในกรณีที่อยากให้ลองทำก่อน 1 รอบ แล้วทำซ้ำไปเรื่อยๆทราบเท่าที่เงื่อนไขเป็นจริง



ฟังก์ชั่น (Function)

ฟังก์ชั่น คืออะไร

ความหมายที่ 1 :

ชุดคำสั่งที่นำมาเขียนรวมกันเป็นกลุ่มเพื่อให้เรียกใช้งานตามวัตถุประสงค์ที่ต้องการ และลดความซ้ำซ้อนของคำสั่งที่ใช้งานบ่อยๆ ฟังก์ชั่นสามารถนำไปใช้งานได้ทุกที่และแก้ไขได้ในภายหลัง ทำให้โค้ดในโปรแกรมมีระเบียบและใช้งานได้สะดวกมากยิ่งขึ้น

ความหมายที่ 2 :

โปรแกรมย่อยที่นำเข้ามาเป็นส่วนหนึ่งของโปรแกรมหลัก เพื่อให้สามารถเรียกใช้งานได้โดยไม่จำเป็นต้องเขียนโค้ดคำสั่งใหม่ทั้งหมด

ชนิดของฟังก์ชั่น

- **ฟังก์ชั่นมาตรฐาน (Built-In Function)** คือ ฟังก์ชั่นที่มีอยู่ในภาษา PHP ผู้ใช้สามารถเรียกใช้งานได้เลย
- **ฟังก์ชั่นที่ผู้ใช้สร้างขึ้นมาเอง (User-Define Function)** คือ ฟังก์ชั่นที่ถูกสร้างขึ้นมาเพื่อวัตถุประสงค์ให้ทำงานตามที่ผู้ใช้ต้องการ

กฎการตั้งชื่อฟังก์ชั่น

- ชื่อฟังก์ชันต้องไม่ซ้ำกัน
- ชื่อฟังก์ชันสามารถเป็นตัวอักษร ตัวเลข หรือขีดเส้นล่าง (underscores)
- ชื่อของฟังก์ชันต้องไม่ขึ้นต้นด้วยตัวเลข

รูปแบบของฟังก์ชั่น

1.ฟังก์ชั่นที่ไม่มีการรับและส่งค่า

```
function ชื่อฟังก์ชั่น(){
    // คำสั่งต่างๆ
}
```

การเรียกใช้งานฟังก์ชั่น

```
ชื่อฟังก์ชั่น ();
```

รูปแบบของฟังก์ชั่น

2.ฟังก์ชั่นที่มีการรับค่าเข้ามาทำงาน

```
function ชื่อฟังก์ชั่น(parameter1,parameter2,.....){

// กลุ่มคำสั่งต่างๆ

}
```

อาร์กิวเมนต์ คือ ตัวแปรหรือค่าที่ต้องการส่งมาให้กับฟังก์ชัน (ตัวแปรส่ง)
พารามิเตอร์ คือ ตัวแปรที่ฟังก์ชันสร้างไว้สำหรับรับค่าที่จะส่งเข้ามาให้กับฟังก์ชัน (ตัวแปรรับ)

การเรียกใช้งานฟังก์ชั่น

```
ชื่อฟังก์ชั่น (argument1,argument2,.....);
```

รูปแบบของฟังก์ชั่น

3.ฟังก์ชั่นที่มีส่งค่าออกมา

```
function ชื่อฟังก์ชั่น(){
    return ค่าที่จะส่งออกไป
}
```

รูปแบบของฟังก์ชั่น

4.ฟังก์ชั่นที่มีการรับค่าเข้ามาและส่งค่าออกไป

```
function ชื่อฟังก์ชั่น(parameter1,parameter2,.....){
    retrun ค่าที่จะส่งออกไป
}
```

ฟังก์ชั่นแบบกำหนดค่าเริ่มต้น

```
function ชื่อฟังก์ชั่น (name=”kongruksiam”,parameter2,.....){
    // คำสั่งต่างๆ
}
```

ขอบเขตตัวแปร

- **local variable** ตัวแปรที่ทำงานอยู่ในฟังก์ชั่นมีขอบเขตการทำงานตั้งแต่จุดเริ่มต้นไปจนถึงจุดสิ้นสุดของฟังก์ชั่น

- **global variable** ตัวแปรที่ทำงานอยู่นอกฟังก์ชั่นมีขอบเขตการทำงานตั้งแต่จุดเริ่มต้นไปจนถึงจุดสิ้นสุดของไฟล์ที่ประกาศใช้

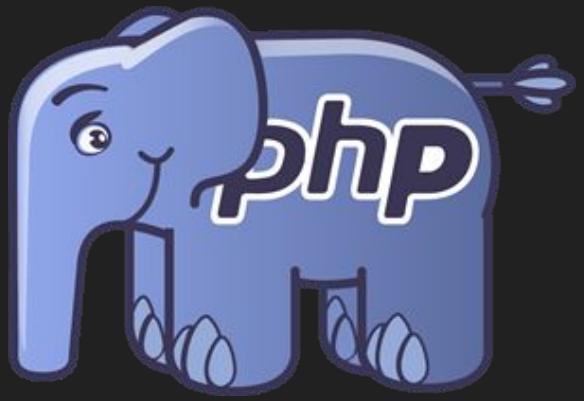
$GLOBALS Keyword

local variable ตัวแปรที่ทำงานอยู่ในฟังก์ชั่นมีขอบเขตการทำงานตั้งแต่จุดเริ่มต้นไปจนถึงจุดสิ้นสุดของฟังก์ชั่น **ถ้าอยากให้เรียกใช้งานในนอกฟังก์ชั่นได้ ให้ใส่ $GLOBALS / globals ข้างหน้าตัวแปร** เช่น

```
function showname(){

    $GLOBALS[“name”]=”kongruksiam”; หรือ

    global $name=”kongruksiam”;

}

echo $name;
```

เรียกใช้งานตัวแปรและฟังก์ชั่นจากไฟล์ด้านนอก

ฟังก์ชั่น	การทำงาน
include()	ฟังก์ชั่นสำหรับโหลดไฟล์เข้ามาทำงาน โดยจะทำงานตั้งแต่เริ่มต้นจนจบโปรแกรม ถ้าหาไฟล์ไม่เจอจะแจ้งเตือนข้อผิดพลาด (warning) แล้วข้ามไปทำงานส่วนอื่นๆต่อ
include_one()	ทำงานเหมือนฟังก์ชั่น include แต่จะเรียกใช้งานไฟล์แค่ครั้งเดียว
require()	ฟังก์ชั่นสำหรับโหลดไฟล์เข้ามาทำงาน โดยจะทำงานตั้งแต่เริ่มต้นจนจบโปรแกรม ถ้าหาไฟล์ไม่เจอจะแจ้งเตือนข้อผิดพลาดและหยุดทำงานโปรแกรม
require_one()	ทำงานเหมือนฟังก์ชั่น require แต่จะเรียกใช้งานไฟล์แค่ครั้งเดียว



อาร์เรย์ (Array)

ข้อจำกัดของชนิดข้อมูลพื้นฐาน

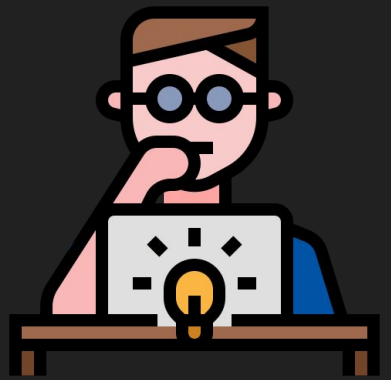
การประกาศตัวแปรแต่ละครั้ง

ตัวแปร 1 ตัวสามารถเก็บข้อมูลได้แค่ 1 ค่าเท่านั้น เช่น

```
$number = 1;
```

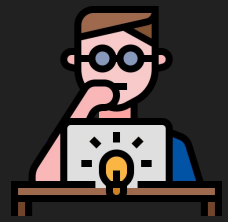
ถ้าอยากเก็บเลข 10 ค่าต้องทำอย่างไร ?

ต้องประกาศตัวแปร 10 ตัวแปร หรือไม่ ?



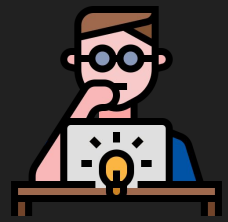
Array คืออะไร

ความหมายที่ 1 ชุดของตัวแปรที่อยู่ในรูปลำดับใช้เก็บค่าข้อมูลให้อยู่ในกลุ่มเดียวกัน ข้อมูลภายในอาร์เรย์จะถูกเก็บบนหน่วยความจำในตำแหน่งที่ต่อเนื่องกัน โดยขนาดของอาร์เรย์จะเล็กหรือใหญ่ขึ้นกับจำนวนมิติที่กำหนดขึ้น



Array คืออะไร

ความหมายที่ 2 เป็นตัวแปรที่ใช้ในการเก็บข้อมูลที่มีลำดับที่ต่อเนื่อง ซึ่งข้อมูลมีค่าได้หลายค่าโดยใช้ชื่ออ้างอิงได้เพียงชื่อเดียว และใช้หมายเลขกำกับ (index) ให้กับตัวแปรเพื่อจำแนกความแตกต่างของค่าตัวแปรแต่ละตัว

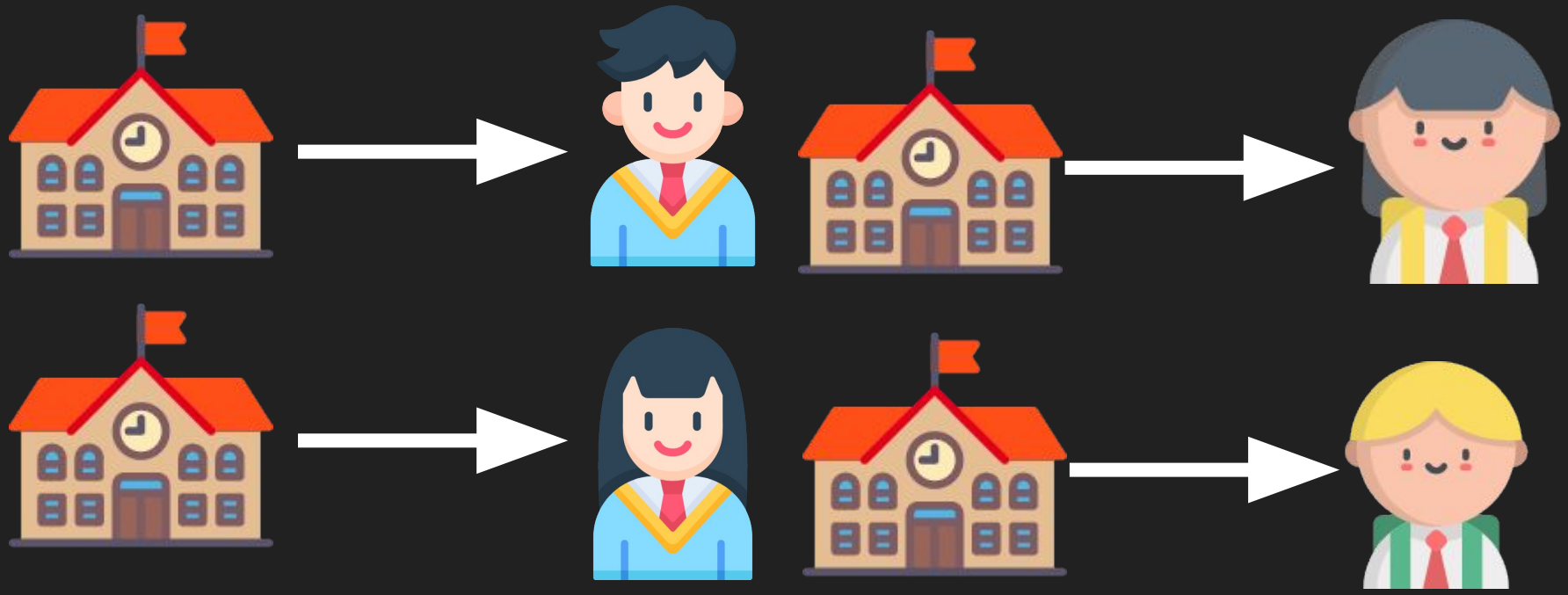


โครงสร้างของ Array

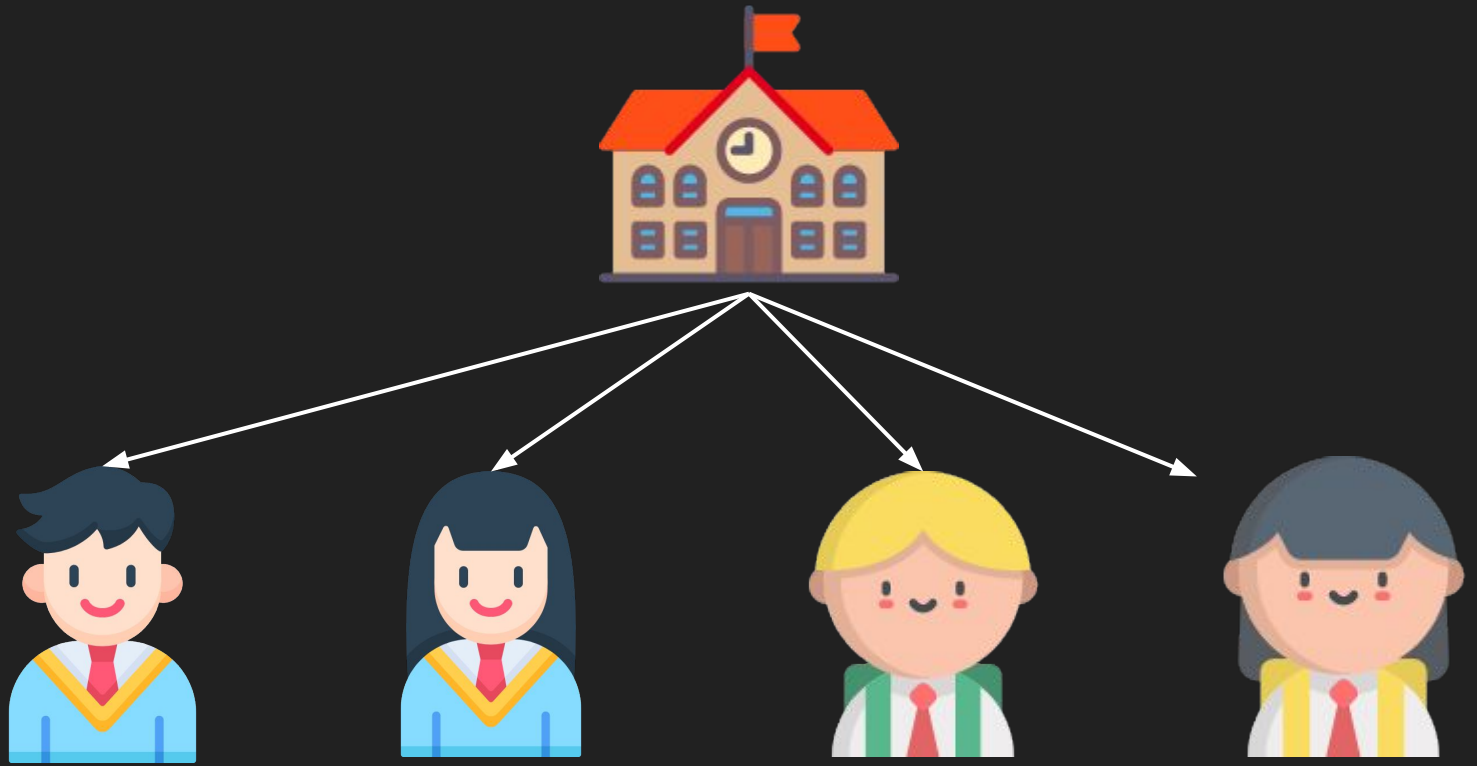
1. ข้อมูลที่อยู่ในอาร์เรย์จะเรียกว่าสมาชิก หรือ อิลิเมนต์ (element)
2. แต่ละอิลิเมนต์ (element) จะเก็บค่าข้อมูล (value) และอินเด็กซ์ (Index) เอาไว้
3. Index หมายถึงคีย์ของอาร์เรย์ใช้อ้างอิงตำแหน่งของ element
4. Index เป็นได้ทั้งตัวเลขหรือตัวอักษร (keys)
5. สมาชิกใน array ต้องมีชนิดข้อมูลเหมือนกัน
6. สมาชิกใน array จะถูกคั่นด้วยเครื่องหมาย comma

ตัวแปร = โรงเรียน

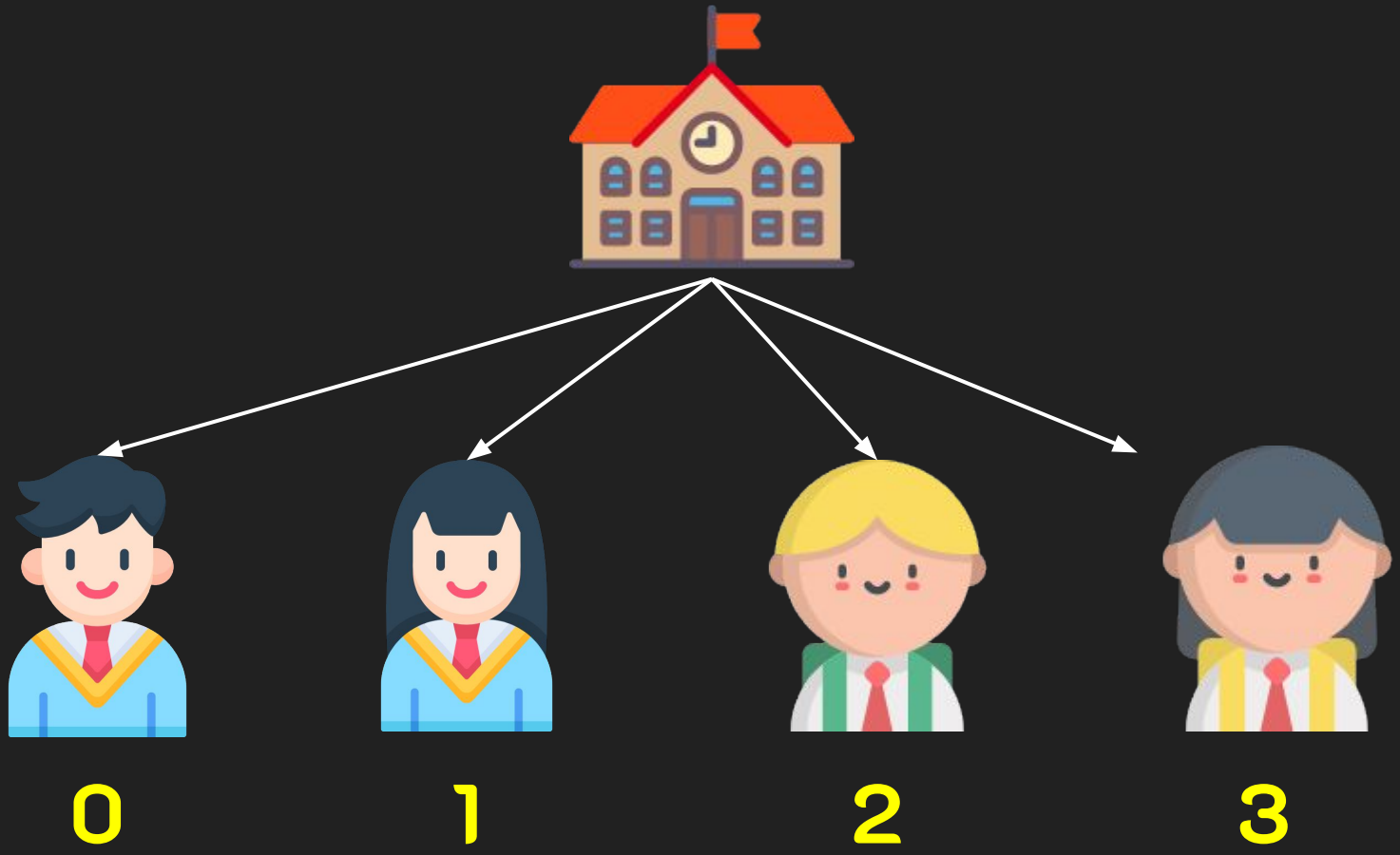
ค่าที่เก็บในตัวแปร = นักเรียน



การสร้างตัวแปรแบบปกติ



การสร้างตัวแปรแบบ Array


0
1
2
3

ประเภทของ Array

1. Array แบบเดี่ยว
2. Array แบบจับคู่ (Associate)

รูปแบบการสร้าง Array

1. ใช้ฟังก์ชั่น array
2. ใช้ฟังก์ชั่น range
3. ใช้สัญลักษณ์กล้ามปู [ ]

การสร้าง Array แบบเดี่ยวโดยใช้ฟังก์ชั่น array

ไม่มีการกำหนดเลข index

- $pets= array(“แมว”,”สุนัข”,”กระต่าย”);

แมว	สุนัข	กระต่าย

- $province=array(“bangkok”,”chonburi”);

bangkok	chonburi

print_r($pets);

การสร้าง Array แบบจับคู่ (Associate)

มีการกำหนด Index (key)

- $pets= array(“cat”=>“แมว”,”dog”=>”สุนัข”,”rabbit”=>”กระต่าย”);
- $province=array(“bangkok”=>36,”chonburi”=>11);
- print_r($pets);

การสร้าง Array โดยใช้ฟังก์ชั่น range

ใช้สร้าง array แบบระบุช่วงตัวเลขหรือตัวอักษร เรียงจากน้อยไปมาก

- range(start,stop,step);
- $number=range(4,8); // ข้อมูลอยู่ในช่วง 4-8
- $number=range(0,20,5); // ข้อมูลอยู่ในช่วง 0-20 ค่าความต่างคือ 5
- $alphabet=range(A,F); // ข้อมูลอยู่ในช่วง A-F
- $alphabet=range(A,F,2); // ข้อมูลอยู่ในช่วง A-F ค่าความต่างคือ 2

การสร้าง Array โดยใช้ [ ]

- เป็นการสร้างและกำหนดค่าให้ array โดยตรง
- $colors=[“ขาว”,”แดง”,”เขียว”]

ขาว	แดง	เขียว

- $colors=[“white”=>”ขาว”,”red”=>”แดง”,”green”=>”เขียว”]
- $colors[1]=“น้ำเงิน”; // เปลี่ยนแปลงค่า index ที่ 1 เป็นสีน้ำเงิน

การเข้าถึงสมาชิกใน array

- การเข้าถึงสมาชิกใน array จะใช้ชื่อ array และ index ในการอ้างอิงค่า array
- Index เริ่มต้นที่ตัวเลข 0
- $pets=array(“แมว”,”สุนัข”,”กระต่าย”);
- $pets[0];
- $pets[1];

แบบกำหนดค่าโดยตรง

- $name[“fname”] = “ก้อง”
- $name[“lname”] = “รักสยาม”
- $name[“age”] = 20

การเข้าถึงสมาชิกด้วย For Loop

- count($pets); นับจำนวนสมาชิกในอาร์เรย์ ใช้กับอาร์เรย์ที่มี index เป็นตัวเลขที่เรียงลำดับจากน้อยไปมาก

- array_count_values($pets); ใช้นับความถี่ของข้อมูลใน array ที่มีค่าซ้ำกัน

การเข้าถึงสมาชิกด้วย While Loop

ใช้ฟังก์ชั่น each ในการอ่านค่า array ทั้งข้อมูลและอินเด็กซ์

การเข้าถึงสมาชิกด้วยฟังก์ชั่น list

ใช้ฟังก์ชั่น list ในการอ่านค่า array ทั้งข้อมูลและอินเด็กซ์และเก็บลงในตัวแปร 2 ตัวแปรโดยการเก็บแยกเป็นข้อมูลและอินเด็กซ์ลงในตัวแปรเหล่านั้น

การเข้าถึงสมาชิกด้วย foreach แบบธรรมดา

```
foreach($array as $value)
        statement
```

```
$days = array(“อาทิตย์”, “จันทร์”, ”อังคาร”, “พุธ”, “พฤหัสบดี”, “ศุกร์”, “เสาร์”);

foreach($days as $value) {
    echo $value . “<br />”;
}
```

การเข้าถึงสมาชิกด้วย foreach แบบจับคู่

```
foreach($array as $key => value)
        statement
```

```
foreach($colors as $key => $val) {
        echo “$key = $val <br />”;
  }
```

Array 2 มิติ

- Array ที่มีข้อมูลสมาชิกภายในเป็น array (array ซ้อน array) เปรียบเสมือนกัน matrix
- มีโครงสร้างเป็นรูปแบบ แถว (แนวนอน) และคอลัมน์ (แนวตั้ง)

Array 2 มิติ

EN	TH	PRICE
Keyboard	คีย์บอร์ด	1500
Mouse	เมาส์	900
Speaker	ลำโพง	2500

Array 2 มิติ

```
$products = array(
    array(“keyboard”,”คีย์บอร์ด”,1500),
    array(“mouse”,”เมาส์”,900),
    array(“speaker”,”ลำโพง”,2500)
)
```

ฟังก์ชั่นใน Array

- ฟังก์ชั่นเรียงลำดับ
- ฟังก์ชั่นเพิ่มและลบสมาชิก Array
- ฟังก์ชั่นสลับค่าข้อมูล
- ตัวชี้ตำแหน่ง Array
- ฟังก์ชั่นเกี่ยวกับ Index และ Value
- การค้นหาข้อมูลใน Array
- การรวม Array

ฟังก์ชั่นเพิ่มและลบสมาชิกใน Array

- **array_push()** - ใช้เพิ่มสมาชิกในตำแหน่งสุดท้าย
- **array_pop()** - ใช้ลบสมาชิกตำแหน่งสุดท้าย
- **array_unshift()** - ใช้เพิ่มสมาชิกในตำแหน่งแรก
- **array_shift()** - ใช้ลบสมาชิกในตำแหน่งแรก
- **array_splice()** - ใช้ลบและเพิ่มสมาชิกในตำแหน่งที่ต้องการ

ฟังก์ชั่นเรียงลำดับ Array แบบเดี่ยว

- **sort()** - เรียงข้อมูลเลขจากน้อยไปมาก
- **rsort()** - เรียงข้อมูลเลขจากมากไปน้อย

- **sort()** - ถ้าเป็นข้อความจะเรียงจากพยัญชนะไปสระ
- **rsort()** - ถ้าเป็นข้อความจะเรียงจากสระไปพยัญชนะ

ฟังก์ชั่นเรียงลำดับ Array แบบจับคู่

- **asort()** - เรียงข้อมูลจากน้อยไปมาก
- **arsort()** - เรียงข้อมูลจากมากไปน้อย
- **ksort()** - เรียง Index จากน้อยไปมาก
- **krsort()** - เรียง Index จากมากไปน้อย

ฟังก์ชั่นสลับค่าใน Array

- **shuffle()** - สุ่มสลับค่าข้อมูล โดยฟังก์ชั่นจะกำหนดค่า index ใหม่และเก็บลงในตัวแปร Array เดิม

- **reverse()** - คัดลอก Array ไปไว้ใน Array ใหม่พร้อมเรียงลำดับแบบย้อนหลับ (Reverse)

ฟังก์ชั่นเกี่ยวกับ Index และ value

- **array_keys()** - ดึง index ทั้งหมดของ Array
- **array_values()** - ดึง value ทั้งหมดของ Array
- **array_flip()** - สลับ index กับ value
- **array_unique()** - ลบค่าข้อมูลซ้ำใน Array

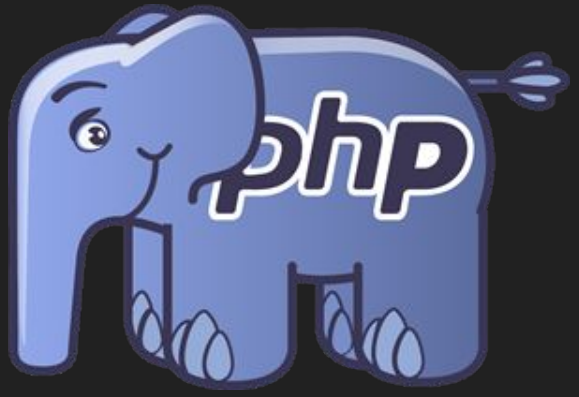
ฟังก์ชั่นการค้นหาข้อมูลใน Array

- **array_key_exists()** - ตรวจสอบว่ามี Index นี้ใน Array หรือไม่ ?

- **in_array()** - ตรวจสอบว่ามี value นี้ใน Array หรือไม่ ?

ฟังก์ชั่นการรวม Array

- **array_merge()** - รวม Array หากมี Index ที่ซ้ำกันจะนำค่าข้อมูล Array ชุดหลังมาทับข้อมูล Array ชุดแรก
- **array_merge_recursive()** - รวม Array หากมี Index ที่ซ้ำกันจะนำค่าข้อมูล Array ชุดหลังมาต่อท้ายข้อมูล Array ชุดแรก
- **array_combine()** - รวม Array โดยใช้ Array ตัวแรกเป็น index และใช้ Array ตัวที่สองเป็น Value



จัดการสตริง(String)

String คืออะไร

ชุดของตัวอักษร ซึ่งเกิดจากการรวมตัวอักษรหลายๆตัวใน PHP โดยในหัวข้อนี้จะมาทำความรู้จักกับฟังก์ชั่นที่จัดการ String ใน PHP เช่น หาความยาวของข้อความ การเปรียบเทียบข้อความ เป็นต้น

การแสดงผล String

- **echo (void echo)** คือฟังก์ชั่นที่แสดงผลข้อความกำหนด String กี่ตัวก็ได้ เช่น echo $str1,$str2 เป็นต้น
- **print()** ฟังก์ชั่นที่แสดงผลข้อความกำหนด String ได้ตัวเดียว

ฟังก์ชั่นเปรียบเทียบ String

ชื่อฟังก์ชั่น	การทำงาน	ผลลัพธ์
strcmp()	เทียบความสำคัญของ String ตัวพิมพ์ใหญ่จะมีค่ามากกว่าตัวพิมพ์เล็ก	เท่ากัน 2 ค่า = 0 str1<str2 = <0 str1>str2 = >0
strcasecmp()	คล้าย strcmp() แต่ ตัวพิมพ์ใหญ่จะมีค่าเท่ากับตัวพิมพ์เล็ก	เท่ากัน 2 ค่า = 0 str1<str2 = น้อยกว่า 0 str1>str2 = มากกว่า 0
strspn()	หาจำนวนตัวอักษรใน str1 ที่พบใน str2 ถ้าเหมือนกันจะค้นหาตัวต่อไปจนกว่าจะไม่เจอ	จำนวนตัวอักษรที่พบ
strcspn()	หาจำนวนตัวอักษรใน str1 ที่ไม่พบใน str2	จำนวนตัวอักษรที่ไม่พบ

หาความยาวของ String

strlen(string)

การตัดช่องว่างของ String

ชื่อฟังก์ชั่น	การทำงาน
trim()	ตัดช่องว่างซ้าย - ขวา ของข้อความ
ltrim()	ตัดช่องว่างด้านซ้ายมือ
rtrim()	ตัดช่องว่างด้านขวามือ

การเปลี่ยนรูปแบบตัวอักษร

ชื่อฟังก์ชั่น	การทำงาน
strtolower()	เปลี่ยนตัวอักษรทุกตัวเป็นตัวพิมพ์เล็ก
strtoupper()	เปลี่ยนตัวอักษรทุกตัวเป็นตัวพิมพ์ใหญ่
ucwords()	เปลี่ยนตัวอักษรตัวแรกของแต่ละคำเป็นตัวพิมพ์ใหญ่
ucfirst()	เปลี่ยนตัวอักษรตัวแรกของข้อความเป็นตัวพิมพ์ใหญ่

การรวมและการแยก String

ชื่อฟังก์ชั่น	การทำงาน
implode()	รวม String เข้าด้วยกันโดยกำหนดอักขระสำหรับรวม String
explode()	แยก String โดยใช้ตัวอักขระเป็นตัวแบ่งและแยกเก็บในรูปแบบ Array
substr()	ใช้ตัดตัวอักษรที่ไม่ต้องการออกจาก String โดยการระบุตำแหน่ง
str_split()	ใช้แยก String ตามความยาวที่กำหนด (Default = 1) แล้วแยกเก็บในรูปแบบของ Array

การค้นหาตัวอักษรใน String

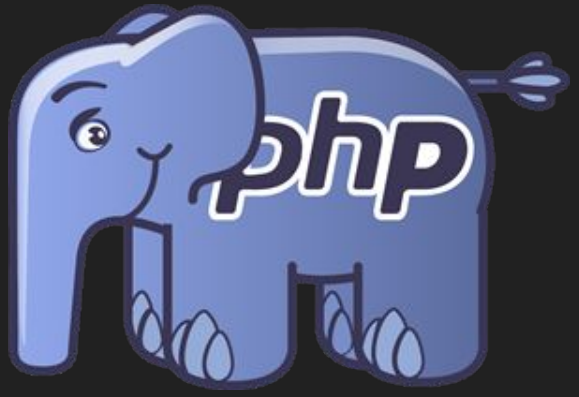
ชื่อฟังก์ชั่น	การทำงาน
strstr()	ค้นหาตัวอักษรหรือ String ย่อย ตัวพิมพ์เล็ก - ตัวพิมพ์ใหญ่จะมีความหมายที่แตกต่างกัน
stristr()	ค้นหาตัวอักษรหรือ String ย่อย ตัวพิมพ์เล็ก - ตัวพิมพ์ใหญ่จะมีความหมายเหมือนกัน

การเปลี่ยนค่า String

ชื่อฟังก์ชั่น	การทำงาน
str_replace()	เปลี่ยน string ย่อยที่พบเป็น string ย่อยใหม่
strrev()	กลับค่า string (reversing string)
str_repeat()	การทำซ้ำ string

การเข้าและถอดรหัส String

ชื่อฟังก์ชั่น	การทำงาน
md5() / sha1()	เข้ารหัสโดยคำนวณ String และส่งค่า hash กลับมา
crypt()	เข้ารหัสโดยสุ่มเวลามาทำงานร่วมด้วย
base64_encode()	เข้ารหัส string ด้วย base_64
base64_decode()	ถอดรหัส string ด้วย base_64



จัดการตัวเลขและเวลา

การจัดการตัวเลข (ปัดเศษทศนิยม)

ฟังก์ชั่น	การทำงาน
ceil ()	ปัดเศษทศนิยมขึ้นทุกกรณี (float)
floor()	ปัดเศษทศนิยมทิ้งทุกกรณี คืนค่าเป็นจำนวนเต็มแบบ float
round()	ปัดเศษทศนิยม >=5 เป็นต้นไปปัดเศษขึ้น ถ้า <5 ให้ปัดเศษลง

การจัดรูปแบบตัวเลข

ฟังก์ชั่น	การทำงาน
number_format()	จัดรูปแบบการแสดงผลตัวเลข เช่น $number=35000
	number_format($number)// 35,000
	number_format($number,2) // 35,000.00

การจัดการเกี่ยวกับวันที่และเวลา

ฟังก์ชั่น	การทำงาน
time()	แสดงวันที่และเวลาปัจจุบันในรูปแบบ Unix Timestamp แสดงเป็นหน่วยวินาที
date()	แสดงวันที่และเวลาโดยการกำหนดรูปแบบลงในฟังก์ชั่นผ่านพารามิเตอร์

รูปแบบวันที่และเวลา

พารามิเตอร์	การทำงาน
d/j	d = วันที่ 2 หลัก (01 - 31) j = วันที่ 1 หลัก (1-31)
D/l	D = แสดงวันแบบตัวย่อ 3 ตัวอักษร (sun , mon) l = แสดงวันแบบเต็ม
W	แสดงลำดับสัปดาห์ของปี 0 - 52 (1 ปีมี 53 สัปดาห์)

รูปแบบวันที่และเวลา

พารามิเตอร์	การทำงาน
F/M	F = แสดงชื่อเดือนแบบเต็ม M = แสดงชื่อเดือนแบบย่อ 3 ตัวอักษร
m/n	m = แสดงชื่อเดือนด้วยตัวเลข 2 หลัก (01 - 12) n = แสดงชื่อเดือนด้วยตัวเลข 1 หลัก (1 - 12)
t	แสดงจำนวนวันของเดือน เช่น 28 - 31

รูปแบบวันที่และเวลา

พารามิเตอร์	การทำงาน
y/Y	y = แสดงปี ค.ศ เลขท้าย 2 หลัก Y = แสดงปี ค.ศ แบบเต็ม
a/A	แสดงเวลาช่วงเช้าหรือช่วงบ่าย a = am หรือ pm A = AM หรือ PM

รูปแบบวันที่และเวลา

พารามิเตอร์	การทำงาน
g/h	g = แสดงชั่วโมงในรูปแบบ 12 ชั่วโมง (1-12) h = แสดงชั่วโมงในรูปแบบ 12 ชั่วโมง (01-12)
G/H	G = แสดงชั่วโมงในรูปแบบ 24 ชั่วโมง (0-23) H = แสดงชั่วโมงในรูปแบบ 24 ชั่วโมง (00-23)
i/s	i = แสดงนาที s = แสดงวินาที

รูปแบบวันที่และเวลา

พารามิเตอร์	การทำงาน
c	วันที่และเวลาแบบตามมาตรฐาน ISO-8601
r	รูปแบบวันที่ในอีเมล RFC 2822

getdate()

พารามิเตอร์	การทำงาน
hours/minutes/second	ชั่วโมง / นาที / วินาที
mday/wday/mon /year/yday	วันที่ / วันในแต่ละสัปดาห์ / ชื่อเดือน /ปี/วันในแต่ละปีเป็นตัวเลข
weekday/month	ชื่อวันในแต่ละสัปดาห์ / ชื่อเดือนแบบเต็ม

checkdate() – ตรวจสอบความถูกต้องของวัน

พารามิเตอร์	การทำงาน
int month/day/year	เดือน / วันที่ / ปี
bool checkdate(int month,int day,int year)