

# การสร้างเว็บเพจด้วย PHP

โดย  
อ.ชูศักดิ์ ยาทองไชย

1

## PHP ()

### 1. แนะนำ PHP

PHP คืออะไร, PHP ทำงานอย่างไร, ทำไมต้องใช้ PHP

### 2. PHP Syntax

รูปแบบและโครงสร้างของ PHP

### 3. PHP Variables

การกำหนด และใช้ตัวแปร, Operator, การกำหนดเงื่อนไข และ Loop

### 4. PHP Function

การเรียกใช้ Functions

### 5. PHP Forms

การรับข้อมูลจาก Forms

2

## PHP ()

### 6. PHP Cookies

Cookie คืออะไร, การกำหนดและการใช้งาน Cookie

### 7. PHP Session

Session คืออะไร, การกำหนดและการใช้ตัวแปรของ Session

### 8. PHP Class และ Object

Object ต่าง ๆ ของ PHP, การกำหนด Properties ของ Object

### 9. PHP Components

Component ต่าง ๆ ที่ใช้กับ PHP

3

## PHP คืออะไร?

- PHP เป็นภาษำพวก scripting language คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปต์ก็เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML
- PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้สามารถสร้างเอกสารแบบ Dynamic HTML

4

## ไฟล์ PHP มีรูปแบบอย่างไร?

- ไฟล์ PHP จะมีลักษณะเหมือนกับไฟล์ HTML
- ไฟล์ PHP จะประกอบด้วย Text, HTML tags และ Scripts
- Scripts ในไฟล์ PHP จะถูกประมวลผลบนฝั่ง Server
- ไฟล์ PHP จะต้องมีส่วนขยายเป็น “.php”

5

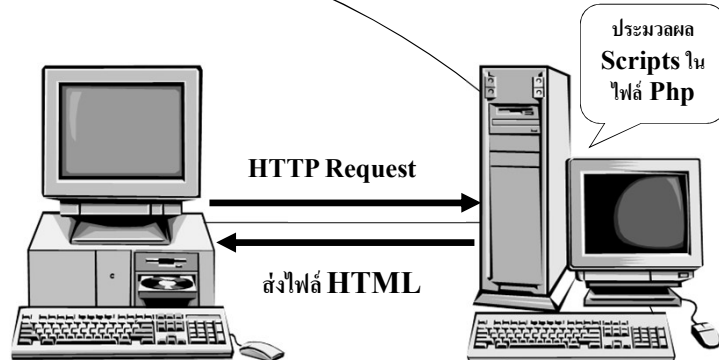
## PHP ทำงานอย่างไร?

การทำงานของ PHP จะใช้ Scripts อยู่ 2 รูปแบบ คือ

- **Server-Side Script** เป็นลักษณะการทำงานบนเครื่อง Server และแปลออกมาเป็นภาษา HTML เช่น ASP, CGI
- **Client-Side Script** เป็นลักษณะการทำงานบนเครื่อง Client (เครื่องผู้ใช้) เช่น JavaScript, VBScript

6

## การทำงานของ PHP



คอมพิวเตอร์ที่รันโปรแกรม Browser

Server ที่รันโปรแกรม IIS หรือ PWS

7

## ทำไมต้องใช้ PHP (1)

- เป็นภาษาที่มีลักษณะเป็นแบบ Open Source ผู้ใช้สามารถ Download และนำ Source Code ของ PHP ไปใช้โดยไม่เสียค่าใช้จ่าย
- เป็นสคริปต์แบบ Server Side Script ดังนั้นจึงทำงานบนเว็บเซิร์ฟเวอร์ ไม่ส่งผลกับการทำงานบนเครื่อง Client โดย PHP จะอ่านโค้ด และทำงานที่เซิร์ฟเวอร์ จากนั้นจึงส่งผลลัพธ์ที่ได้จากการประมวลผลมาที่เครื่องของผู้ใช้ในรูปแบบของเอกสาร HTML ซึ่งโค้ดของ PHP นี้ผู้ใช้จะไม่สามารถมองเห็นได้
- PHP สามารถทำงานในเว็บเซิร์ฟเวอร์หลายชนิด เช่น Personal Web Server (PWS,) Apache, OmniHttpd และ Internet Information Service (IIS)

8

## ทำไมต้องใช้ PHP (2)

- PHP สามารถทำงานได้ในระบบปฏิบัติการที่ต่างชนิดกัน เช่น Unix, Windows, Mac OS หรือ Risc OS อย่างมีประสิทธิภาพ
- PHP มีความสามารถในการทำงานร่วมกับระบบการจัดการฐานข้อมูลที่หลากหลาย ซึ่งระบบการจัดการฐานข้อมูลที่สนับสนุนการทำงานของ PHP เช่น Oracle, FilePro, Solid, FrontBase, MS SQL, mSQL และ MySQL
- PHP อนุญาตให้ผู้ใช้สร้างเว็บไซต์ซึ่งทำงานผ่าน โพรโตคอลชนิดต่างๆ ได้ เช่น LDAP, IMAP, SNMP, POP3 และ HTTP

9

## รูปแบบและโครงสร้างของ PHP (1)

รูปแบบของ PHP มี 4 แบบดังนี้

- Short Style

```
<?echo "My program PHP.<BR>";?>
```

รูปแบบนี้จะใช้ตามรูปแบบของ SGML (Standard Genralized Markup - Language)

- XML Style

```
<?php echo "My program PHP.<BR>";?>
```

รูปแบบนี้สามารถใช้ได้กับภาษา XML (Extensible Markup Language)

10

## รูปแบบและโครงสร้างของ PHP (2)

- SCRIPT Style

```
<SCRIPT LANGUAGE = 'php'> echo "My program PHP.<BR>" ;
</SCRIPT>
```

ใช้ร่วมกับภาษา HTML แล้วระบุภาษาลงไปใน SCRIPT ซึ่ง เหมือนกับการใช้ JavaScript และ VBScript

- ASP Style

```
<% echo "My program PHP.<BR>" ;%>
```

ใช้รูปแบบของแท็กเหมือนกับภาษา ASP (Active Server Pages)

11

## การกำหนดและใช้ตัวแปรของ PHP (1)

การกำหนดชนิดของตัวแปรในการเขียนโปรแกรมภาษา PHP นั้นอาจจะกำหนดหรือไม่ก็ได้ ถ้าไม่กำหนดจะเรียกตัวแปรนั้นว่าเป็นตัวแปรแบบ "Type Strength" คือ ไม่มีการระบุชนิดของตัวแปร โดยมากจะใช้กับตัวแปรที่มีค่าจากการคำนวณตัวเลข เช่น

```
$Price=150*10;
```

ตัวแปร Price จะถูกกำหนดให้เป็นตัวแปรที่มีชนิดเป็น Integer โดยอัตโนมัติเนื่องจากผลลัพธ์ที่ Price เก็บไว้มีชนิดเป็น Integer

กรณีที่มีการกำหนดชนิดให้กับตัวแปรหลังจากที่ใช้ตัวแปรไปแล้ว หรือเมื่อมีการกระทำบางอย่างแล้วต้องการเปลี่ยนชนิดของตัวแปรเพื่อนำไปใช้ต่อ จะเรียกตัวแปรชนิดนี้ว่าตัวแปรแบบ "Type Casting"

12

## ตัวอย่าง แสดงการเปลี่ยนชนิดของตัวแปรเพื่อนำไปใช้ เพื่อการคำนวณ

1. <? php
2. \$x=70.15; //เป็นตัวแปรชนิด Float
3. \$y=50.35; // เป็นตัวแปรชนิด Integer
4. \$z=\$x+\$y; //ผลลัพธ์ที่ได้จะเป็นชนิด Float เนื่องจากตัวแปร \$x มีชนิดเป็น Float
5. \$a=\$z/40;
6. \$b=(Integer)\$a; //เปลี่ยนชนิดของ \$z จาก Float ให้เป็น Integer เรียกว่า "Type Casting"
7. Echo \$b; //แสดงค่าของ \$b
8. ?>

13

## การกำหนดและใช้ตัวแปรของ PHP (2)

### ตัวแปรและการอ้างอิง

ชื่อตัวแปรในภาษา PHP จะมีลักษณะเช่นเดียวกับชื่อตัวแปรในภาษา Perl กล่าวคือตัวแปรทุกตัว จะขึ้นต้นด้วยเครื่องหมาย "\$" เสมอ นอกจากนี้ตั้งแต่ PHP version 4 เป็นต้นมามีวิธีการ ในการอ้างอิงตัวแปรแบบ reference ได้อีกด้วย ทั้งนี้การอ้างอิงตัวแปรแบบ reference จะมีลักษณะเหมือนกับการอ้างอิงด้วย Pointer ในภาษา C คือข้อมูลจะถูกชี้ไปยังตำแหน่งเดียวกัน ในหน่วยความจำ โดยการอ้างอิงในลักษณะนี้จะใช้อ้างอิงถึงตัวแปรตำแหน่งเดียวกันด้วยชื่อมากกว่า 1 ชื่อ

14

## ตัวอย่าง การอ้างอิงตัวแปรแบบ Reference

```
<?php
$A = "Test";
$B = 10;
$C = 10.01;
$ptr = &$A;
echo ("A : $A\n"); // A : Test
echo ("B : $B\n"); // B : 10
echo ("C : $C\n"); // C : 10.01
echo ("ptr : $ptr\n"); // ptr : Test

$ptr="Hello, $ptr";
$C=$B;
$C=200;
echo ("ptr : $ptr\n"); // ptr : Hello, Test
echo ("B : $B\n"); // B : 10.01
echo ("C : $C\n"); // C : 200
echo ("A : $A\n"); // A : Hello, Test
```

>

15

## ตัวแปร

ในบางกรณีเพื่ออำนวยความสะดวกอาจจะมีการอ้างอิงถึงตัวแปรตัวแปรก็ได้เช่นกัน เช่น กรณีต้องการอ้างอิงถึงตัวแปรชื่อ \$stest อาจทำได้โดยการอ้างอิงถึง \$\$a โดย \$a มีความเป็น "test" ก็ได้

16

### ตัวอย่าง การอ้างอิงตัวแปร

```
<?php
  $a= "test;
  $$a="variable";
  echo ("$a ${$a}"); // ทั้งสองบรรทัดจะได้ผลลัพธ์
  เหมือนกัน
  echo ("$a $test"); //
?>
```

17

### การอ้างอิง Array ด้วยตัวแปร

ผู้ใช้จะต้องเขียนเพื่อแยกแยะความสับสน ของการประมวลผลด้วย เช่น  
 \$\$a[1] อาจมีความหมายได้ว่า \${\$a[1]} หรือ \${\$a}[1] ก็ได้

ตัวอย่าง การอ้างอิง Array ด้วยตัวแปรตัวแปร

```
<?php
  $a="HELLO..";
  $b[1]="Test";
  $Test="5555";
  $x="a";
  echo ${$x}[1]; // จะได้ผลลัพธ์ เป็น "E" ซึ่งเป็นตัวอักษรที่ 2 ใน "A"
  echo ${$b[1]}; // จะได้ผลลัพธ์ เป็น "5555" ซึ่งเป็นค่าของ $Test
?>
```

18

### ตัวแปรจากระบบ

ตัวแปรจากระบบคือตัวแปรที่ถูกกำหนดขึ้นโดย **Web Server** และผ่านต่อมายัง **PHP** ซึ่งรายละเอียดจะแตกต่างกันไป ใน **Web Server** แต่ละตัว ทั้งนี้เราอาจจะตรวจสอบได้ว่า **Web Server** ของเรานั้นมีการส่งค่าใดบ้างมาถึง **PHP** ได้ โดยการเรียกใช้ **Function phpinfo()**; และดูในส่วนของ **Web Server** ของตน (กรณีเป็น **Apache** ก็ให้ดูในส่วน **Apache** กรณีเป็น **IIS** ก็ให้ดูในส่วน **IIS** เป็นต้น )

19

### ตัวแปรจาก PHP

เป็นข้อมูลที่ตัวแปลภาษา PHP เตรียมให้ผู้ใช้ก่อนเริ่มทำการประมวลผลคำสั่งต่างๆ ทั้งนี้ตัวแปรบางตัว จะมีการกำหนดค่าเริ่มต้นโดยตัวแปลภาษา PHP หรือไม่นั้นขึ้นอยู่กับคำสั่งค่าบางตัวใน **PHP.INI** ด้วย

20

## ตัวแปรระบบที่ได้จาก PHP

Variable	Description
argv	เป็น Array ของ argument ที่เรียกเข้าสู่ PHP ใน Command-line หรือกรณี GET Method จะแสดง QUERY_STRING (ในลักษณะเดียวกับภาษา C)
argc	แสดงจำนวนของ argument ที่เรียกเข้าสู่ PHP ใน Command-line
PHP_SELF	แสดงชื่อ SCRIPT ที่กำลังทำการประมวลผล (ไม่สามารถใช้ในการเรียก PHP แบบ Command-line)

21

## ตัวแปรจาก Web (1)

ตัวแปรที่ได้รับจากหน้า Web Page นั้นอาจจะได้รับมาจาก URL หรือ Form ผ่าน Get Method, Post Method หรือ อาจจะมาจก Cookie ก็ได้ ทั้งนี้ใน PHP จะเรียกตัวแปรที่ได้รับผ่าน Web ว่า GPC ซึ่งเป็นตัวย่อของ Get, Post และ Cookie ตามลำดับ ซึ่งโดยปกติ PHP ก็จะประมวลผล ในการตั้งค่าของข้อมูลตามลำดับเหล่านี้ เช่นเดียวกัน ดังนั้นข้อมูลใน Cookie จะมีลำดับความสำคัญสูงสุด และข้อมูลใน Get Method จะมีความสำคัญต่ำสุด (สำหรับผู้ที่ไม่คุ้นเคยกับการทำงานของ Web นั้น Get Method คือการผ่าน Parameter ผ่าน URL เช่น `http://www.test.com/test.php?test=5555` เป็นต้น) กรณีนี้ที่ผู้ใช้ต้องการจะเปลี่ยนลำดับในการประมวลผล ค่าตัวแปรจาก Web นั้น ผู้ใช้อาจจะตั้งค่าที่ `gpc_order` ใน PHP.INI ก็ได้

22

## ตัวแปรจาก Web (2)

การอ้างอิงถึงตัวแปรที่ได้รับผ่าน Web สามารถกระทำได้ 2 วิธี คือ การอ้างอิงผ่าน Global Variable หรือ การอ้างอิงผ่าน Array ชื่อ `HTTP_*_VARS` โดย \* คือแหล่งกำเนิดของข้อมูลเช่น GET,POST,COOKIE,SERVER หรือ ENV เป็นต้น (ในกรณีของ PHP Version 4.0 เป็นต้นไปนั้น ผู้ใช้ต้องตั้งค่า `register_globals=on` ใน PHP.INI ก่อนจึงจะทำการอ้างอิงค่าผ่าน Global Variable ได้) ดังตัวอย่างของ Form ต่อไปนี้

23

## ตัวอย่าง Form

```
<form method=post>
<input type=text name="myvar">
</form>
<?php
echo $myvar;
// or
echo $HTTP_POST_VARS["myvar"];
?>
```

24

## การใช้ IMAGE SUBMIT

ในบางกรณีผู้เขียน Web อาจจะใช้ Image แทนการใช้ปุ่ม Submit ก็ได้ ซึ่งในกรณีนี้ PHP จะกำหนดตัวแปรเพิ่มขึ้นอีก 2 ตัว คือ `*_x` และ `*_y` เพื่อใช้แสดงตำแหน่ง X และ Y ของปุ่มที่ถูกกด

25

## PHP Operator

### ● Arithmetic Operators (เครื่องหมายที่ใช้ในการคำนวณ)

เครื่องหมาย	ชื่อ	ความหมาย	ตัวอย่าง
+	Addition	บวก	<code>\$A + AB</code>
-	Subtraction	ลบ	<code>\$A - AB</code>
*	Multiplication	คูณ	<code>\$A * AB</code>
/	Division	หาร	<code>\$A / AB</code>
%	Modulus	เศษจากการหาร (Mod)	<code>\$A % AB</code>

26

## PHP Conditional Statement (1)

กลุ่มคำสั่งแบบมีเงื่อนไข

เป็นกลุ่มคำสั่งที่ใช้ตัดสินใจในการเลือกเงื่อนไขต่าง ๆ ภายในโปรแกรมเพียงเงื่อนไขเดียวมาทำงาน ได้แก่ คำสั่ง `if` และ `switch`

### ● คำสั่ง `if`

เป็นคำสั่งที่ใช้สำหรับตัดสินใจ โดยถ้าเป็นจริงจะทำตามคำสั่งต่างๆที่กำหนดไว้ภายใต้เงื่อนไข `if` ซึ่งเงื่อนไขของคำสั่ง `if` จะอยู่ภายในเครื่องหมายวงเล็บ ( ) แต่ถ้าไม่จริงจะไปทำงานคำสั่งถัดไปที่อยู่หลังเครื่องหมายปีกกาปิด } มีรูปแบบของคำสั่ง `if` ดังนี้

```
if (เงื่อนไข)
{ คำสั่งต่าง ๆ เมื่อเงื่อนไขเป็นจริง
```

27

## ตัวอย่าง แสดงการใช้คำสั่ง `if`

- `if ($height > 0 && $width > 0)`
- `{ $area = $height * $width;`
- `echo "พื้นที่สี่เหลี่ยมเท่ากับ". $area ;`
- `}`

28

## PHP Conditional Statement (2)

- คำสั่ง if...else

เป็นคำสั่งที่มี 2 เงื่อนไขคือ ถ้าเงื่อนไขภายใน if เป็นจริงก็เข้าไปทำคำสั่งภายใน if แต่ถ้าเงื่อนไขเป็นเท็จก็จะไปทำคำสั่งภายใน else รูปแบบเป็นดังนี้

```
if (เงื่อนไข)
{ คำสั่งต่าง ๆ เมื่อเงื่อนไขเป็นจริง ; }

else
{ คำสั่งต่าง ๆ เมื่อเงื่อนไขเป็นเท็จ ; }

...
```

29

## ตัวอย่าง แสดงการใช้คำสั่ง else

1. if (\$height > 0 && \$width > 0)
2. { \$area = \$height \* \$width;
3. echo "พื้นที่สี่เหลี่ยมเท่ากับ". \$area ;
4. }
5. else
6. {echo "กรุณาป้อนตัวเลขใหม่ <br>;";
7. echo "ค่าข้อมูลมากกว่า 0" ;}

30

## PHP Conditional Statement (3)

- คำสั่ง if...elseif

ใช้ในกรณีที่มีเงื่อนไข if และ else จำนวนมาก คำสั่ง elseif เป็นการรวมกันของคำสั่ง if และ else ซึ่งคำสั่งเหล่านี้จะเรียงลำดับกันอยู่ มีรูปแบบดังนี้

```
if (เงื่อนไขที่ 1)
{ คำสั่งต่าง ๆ เมื่อเงื่อนไขที่ 1 เป็นจริง ; }

elseif (เงื่อนไขที่ 2)
{ คำสั่งต่าง ๆ เมื่อเงื่อนไขที่ 2 เป็นจริง ; }

elseif (เงื่อนไขที่ 3)
{ คำสั่งต่าง ๆ เมื่อเงื่อนไขที่ 3 เป็นจริง ; }

...
```

31

## ตัวอย่าง แสดงการใช้คำสั่ง if...elseif

1. if (\$score < 0)
2. echo "กรุณาตรวจสอบช่วงคะแนนอีกครั้ง";
3. elseif (\$score > 80.0)
4. \$grade = "G";
5. elseif (\$score >= 60.0)
6. \$grade = "P";
7. elseif (\$score < 60)
8. \$grade = "F";

32



## PHP Conditional Statement (4)

- คำสั่ง switch

หลักการการทำงานของ switch จะคล้ายกับคำสั่ง if แต่ว่าเงื่อนไขใน switch จะมีมากกว่า 2 เงื่อนไข คือ ตรวจสอบเงื่อนไขโดยดูจากค่าของตัวแปร โดยนำค่าจากตัวแปรที่อยู่ในคำสั่ง switch มาตรวจสอบกับค่าที่อยู่หลังคำสั่ง case แต่ละคำสั่งถ้าตรงกันจะทำงานตามคำสั่งที่อยู่ใน case นั้น ถ้าไม่ตรงเลยจะมาทำงานที่คำสั่ง default แทน มีรูปแบบดังนี้

33

## รูปแบบของคำสั่ง switch

```
switch (ตัวแปร)
{ case ค่าที่ 1 : คำสั่งที่ 1 ;
      break ;
  case ค่าที่ 2 : คำสั่งที่ 2 ;
      break ;
  ...
  case ค่าที่ N : คำสั่งที่ N ;
      break ;
  default : คำสั่งเมื่อ ไม่มีค่าที่ตรงกับที่ระบุใน case ;
}
```

34

## ตัวอย่าง แสดงการใช้คำสั่ง switch

```
1. Switch ($year)
2. {
3.   Case 5 : $Irate = 12;
4.   break;
5.   Case 15 : $Irate = 18;
6.   break;
7.   Case 30 : $Irate = 24;
8.   break;
9.   Default : echo "ตัวเลือกผิด ป้อนตัวเลข 5 , 15 หรือ 30 เท่านั้น กรุณาป้อนใหม่อีกครั้ง";
10. }
```

35

## PHP Looping Statement (1)

### กลุ่มคำสั่งแบบวนรอบ

เป็นกลุ่มคำสั่งที่ใช้ในการวนรอบ (loop) โปรแกรมจะทำงานไปเรื่อยๆจนกว่าเงื่อนไขที่กำหนดไว้จะเป็นจริงจึงหลุดออกจากการทำงาน คำสั่งในกลุ่มนี้ได้แก่ while, for, do...while

- คำสั่ง while

คำสั่ง while จะคล้ายกับคำสั่ง if มีข้อแตกต่างคือ คำสั่ง if จะทำคำสั่งภายใน if ทีละคำสั่งรอบเดียวหากเงื่อนไขเป็นจริง คำสั่ง while จะทำตามคำสั่งภายใน while ตลอดที่เงื่อนไขเป็นจริง

36

## รูปแบบของคำสั่ง while

**while** (เงื่อนไข)  
 { คำสั่งต่าง ๆ เมื่อเงื่อนไขเป็นจริง; }

ตัวอย่าง

1. \$sum = 100;
2. \$num = 1;
3. While(\$num<=\$sum)
4. {echo \$num. "<BR>";
5. \$num++;
6. }

37

## PHP Looping Statement (2)

- คำสั่ง for

เป็นรูปแบบที่ใช้ในการตรวจสอบเงื่อนไขพร้อมทั้งค่าเริ่มต้นและการเปลี่ยนแปลงของค่าเริ่มต้นไปพร้อม ๆ กัน โดยครบไคที่เงื่อนไขในคำสั่ง for เป็นจริง

มีรูปแบบดังนี้

for (ค่าเริ่มต้นของตัวแปร ; เงื่อนไข ; เปลี่ยนแปลงค่าตัวแปร)  
 { คำสั่งต่างๆ เมื่อเงื่อนไขเป็นจริง ; }

ตัวอย่าง

1. for (\$x = 0;\$x<10 ;\$x++)
2. {echo "รอบที่".\$x;}

38

## PHP Looping Statement (3)

- คำสั่ง do...while

โปรแกรมจะทำงานตามคำสั่งต่างๆ ก่อน เมื่อทำเสร็จจึงจะมาตรวจสอบเงื่อนไขที่คำสั่ง while ถ้าเงื่อนไขเป็นจริงจะวนกลับขึ้นไปทำงานที่คำสั่งต่างๆ ใหม่ แต่ถ้าไม่เป็นจริงจะหลุดออกมาทำงานที่คำสั่งต่อไป มีรูปแบบดังนี้

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

39

## ตัวอย่าง แสดงการใช้คำสั่ง do...while

1. \$sum = 100;
2. \$num = 1;
3. do{
4. echo \$num. "<BR>";
5. \$num++;
6. }
7. While (\$num<=\$sum);

40

## PHP Looping Statement (4)

กลุ่มคำสั่งกระโดดออกจากคำสั่งบรรจบ

เป็นคำสั่งที่ทำงานร่วมกับกลุ่มคำสั่งแบบบรรจบ เพื่อให้หลุดออกจากลูปหรือทำงานต่อไปภายในลูปซึ่งประกอบคำสั่ง **break**, **continue** และ **exit**

- คำสั่ง **break**

ถ้าโปรแกรมพบคำสั่งนี้ จะหลุดจากการทำงานในลูปทันทีเพื่อไปทำคำสั่งอื่นที่อยู่นอกลูป

- คำสั่ง **continue**

คำสั่งนี้จะทำให้หยุดการทำงาน แล้วย้อนกลับไปเริ่มต้นการทำงานที่ลูปใหม่ทำงานคล้ายๆกับ break แต่แตกต่างกันที่ break ใช้หยุดการทำงานในลูปใหม่ทั้งหมด แต่ continue จะข้ามคำสั่งต่างๆแล้วย้อนกลับไปเริ่มต้นลูปใหม่

- คำสั่ง **exit**

ใช้คำสั่ง exit เพื่อให้โปรแกรมหยุดทำงาน กรณีที่ต้องการตรวจสอบข้อผิดพลาดในโปรแกรมและออกจากโปรแกรมการทำงานทั้งหมด

41

## ตัวอย่าง แสดงการใช้คำสั่ง break, continue, exit

```

1. $num = 0;
2. while(1<2)
3. {
4.     $num = $num + 1;
5.     if ($num == 10)
6.     { echo "จบการทำงานใน while loop<BR>";
7.       break;}
8.     if ($num > 0 && $num <= 9)
9.     { echo "ตัวเลข". $num. "<BR>";
10.      continue;}
11. }
12. exit;
13. Echo "โปรแกรมการใช้คำสั่งกระโดดออกจากลูป"
```

42

## PHP Functions (1)

- ในการใช้งาน PHP นอกเหนือจากการควบคุมโปรแกรมให้เป็นไปตามที่ต้องการแล้ว ยังมีอีกสิ่งหนึ่งที่มีความสำคัญและจำเป็นในการเขียนโปรแกรม เพื่อช่วยให้การทำงานมีประสิทธิภาพสูงมากขึ้นและทำงานได้ดีขึ้น คือ "ฟังก์ชัน (Function)"

- ฟังก์ชันคือโปรแกรมย่อยที่สามารถทำการประมวลผลได้ด้วยตัวเอง และสามารถคืนผลลัพธ์ จากการประมวลนั้นกลับสู่โปรแกรมหลัก ที่เรียกใช้ฟังก์ชันนั้น ฟังก์ชันของ PHP มี 2 ส่วนด้วยกันคือ

1. ฟังก์ชันที่ผู้ใช้สร้างเอง
2. ฟังก์ชันที่มีกับ PHP ซึ่งสามารถเรียกใช้งานได้เลย

43

## PHP Functions (2)

### ฟังก์ชันที่ผู้ใช้สร้างเอง

PHP มีความยืดหยุ่นให้ผู้ใช้สามารถสร้างฟังก์ชันขึ้นมาใช้งานได้เองโดยไม่ต้องคอมไพล์ การสร้างก็ทำได้โดยอาศัยโครงสร้างพื้นฐาน, ตัวแปร, ค่าคงที่, โอเปอเรเตอร์ และการควบคุม โปรแกรม สำหรับฟังก์ชันที่ผู้ใช้สร้างเอง จะมี 2 แบบคือ

1. ฟังก์ชันที่ไม่มีการส่งค่าระหว่างฟังก์ชัน
2. ฟังก์ชันที่มีการส่งค่าระหว่างฟังก์ชัน

44

## PHP Functions (3)

### ฟังก์ชันที่ไม่มีการส่งค่าระหว่างฟังก์ชัน

ฟังก์ชันแบบนี้ผู้ใช้จะต้องกำหนดชื่อและขั้นตอนการทำงานของฟังก์ชันไว้ที่ต้นของโปรแกรมก่อน หลังจากนั้นจึงจะสามารถเรียกใช้งานได้ด้วยรูปแบบดังนี้

```
Function Func_Name();
{
    Instructions;
}
Func_Name(); // เรียกใช้งาน
```

45

## ตัวอย่าง การสร้างฟังก์ชันที่ไม่มีการส่งค่าระหว่างฟังก์ชัน

```
<?php
Function Welcome()
{
    echo"ยินดีต้อนรับผู้เยี่ยมชมของเรา."<br>;
}
Welcome(); // เรียกใช้งาน
?>
```

46

## PHP Functions (4)

### ฟังก์ชันที่มีการส่งค่าระหว่างฟังก์ชัน

ฟังก์ชันในลักษณะนี้จะเป็นการส่งค่าเพื่อไปคำนวณจนได้ผลลัพธ์แล้วจะส่งค่ากลับออกมา ภายในฟังก์ชันจะเป็นการใช้โครงสร้าง ตัวแปร , ค่าคงที่, โอเปอเรเตอร์ และอื่นๆ เมื่อกำหนดฟังก์ชันเรียบร้อยแล้ว หลังจากนั้นก็สามารถใช้งานได้ด้วยการส่งค่าเข้าไป โดยรูปแบบจะเป็นดังนี้

```
Function Func_Name(parameter);
{
    return [Instructions];
}
Func_Name(parameter_value);
```

47

## ตัวอย่าง การสร้างฟังก์ชันที่มีการส่งค่าระหว่างฟังก์ชัน

```
<?php
Function Dec_to_Bin($dec) //แปลงเลขฐาน 10 เป็นฐาน 2
{
    return DecBin($dec); //ใช้ฟังก์ชันทางคณิตศาสตร์เข้ามาช่วย
}
Function Bin_to_Dec($bin) //แปลงเลขฐาน 2 เป็นฐาน 10
{
    return BinDec($bin); //ใช้ฟังก์ชันทางคณิตศาสตร์เข้ามาช่วย
}
echo"เลข 2638961 มีค่า = ".Dec_to_Bin(2638961)." ในเลขฐาน 2<br>";
$Diaw = Dec_to_Bin(2638961); //นำค่าที่ได้จากฟังก์ชันแรกมาเก็บในตัวแปร แล้วคำนวณกลับ
echo"เลข ".$Diaw." มีค่า = ".Bin_to_Dec($Diaw)." ในเลขฐาน 10<br>";
?>
```

48

## String Functions (1)

- addslashes() จะทำการเพิ่มเครื่องหมาย \ (Back Slashes) ให้กับ String ที่มีสัญลักษณ์ (', ", \)
- chr() เป็นฟังก์ชันในการแปลงค่าตัวเลข ASCII กลับมาเป็นตัวอักษร
- echo() ใช้ในการแสดงข้อความ สามารถแสดงได้หลายบรรทัด และสามารถใช้รหัสครอบ String ได้ด้วย
- explode() ใช้ในการแบ่ง String โดยต้องกำหนดอักขระที่ใช้แบ่ง เมื่อแบ่งเสร็จแล้วค่า String ที่ถูกแบ่งย่อยจะเก็บอยู่ในตัวแปรอาร์เรย์
- htmlspecialchars() ใช้ในกรณีที่ต้องการแสดงแท็ก HTML บน Browser
- print() ใช้ในการแสดงข้อความคล้ายกับคำสั่ง echo()
- sprintf() ใช้ในการแสดงข้อความแต่สามารถกำหนดรูปแบบได้
- strlen() ใช้ในการคำนวณอักขระทั้งหมดใน String

49

## String Functions (2)

- strpos() ใช้ในการหาตำแหน่งของ String โดยที่ตำแหน่งแรกคือ 0
- strrev() ใช้ในการสลับตำแหน่ง String จากหลังมาหน้า
- strstr() ใช้ในการค้นหาค่าใน String
- strtolower() ใช้ในการเปลี่ยน String ให้เป็นตัวพิมพ์เล็กทั้งหมด
- strtoupper() ใช้ในการเปลี่ยน String ให้เป็นตัวพิมพ์ใหญ่ทั้งหมด
- str\_replace() ใช้ในการค้นหาและแทนที่ String
- trim() ใช้ในการตัดช่องว่างทั้งหน้าและหลัง String
- ucfirst() ใช้ในการเปลี่ยนตัวอักษรตัวแรกให้เป็นตัวพิมพ์ใหญ่
- ucwords() ใช้ในการเปลี่ยนตัวอักษรตัวแรกของทุกคำให้เป็นตัวพิมพ์ใหญ่

50

## Mail Functions

เป็นฟังก์ชันที่ใช้ในการติดต่อกับระบบ Mail ซึ่งมีรูปแบบในการใช้งานที่ดังตัวอย่างต่อไปนี้

```
<?
$Address = "aspthai@aspthai.net";
$Subject = "Test PHP Mail Function";
$message = "I Love You";
$From = "diaw@aspthai.net";

mail($Address,$Message,"From : $From");
echo "Send Mail Successfully";
?>
```

51

## Data Files functions (เพิ่มข้อมูล Text Files) (1)

- fopen() ใช้ในการเปิดเพิ่มข้อมูลและสร้างเพิ่มข้อมูล ซึ่งมีโหมดในการทำงานดังนี้
  - r เปิดเพิ่มข้อมูลสำหรับการอ่านข้อมูล
  - r+ เปิดเพิ่มข้อมูลสำหรับการอ่านข้อมูล / บันทึกข้อมูล
  - w สร้างเพิ่มข้อมูลสำหรับบันทึกข้อมูล
  - w+ สร้างเพิ่มข้อมูลสำหรับการอ่าน / บันทึกข้อมูล
  - a ใช้สำหรับบันทึกข้อมูลต่อจากเพิ่มข้อมูลเดิม
  - a+ ใช้ต่อหรือสร้างข้อมูลเพิ่มเข้าไปในเพิ่มข้อมูลเดิม

52

## Data Files functions (แฟ้มข้อมูล Text Files) (2)

- **fclose()** ใช้ในการปิดแฟ้มข้อมูล
- **fputs()** ใช้ในการบันทึกข้อมูล ตรงตัวชี้ตำแหน่งของข้อมูล
- **fgets()** ใช้ในการอ่านข้อมูล ตรงตัวชี้ตำแหน่งของข้อมูล
- **feof()** ใช้ในการตรวจสอบแฟ้มข้อมูล โดยในการตรวจสอบนั้น ถ้ายังไม่จบแฟ้มข้อมูล ฟังก์ชันนี้จะคืนค่าเป็น 0 แต่ถ้าจบแฟ้มข้อมูลไปแล้ว ฟังก์ชันนี้จะคืนค่าที่ไม่ใช่ 0 ออกมา

53

## Mathematical Functions (1)

- **sin()** หาค่า Sine ทางตรีโกณมิติของมุมที่ต้องการ
- **cos()** หาค่า Cosine ทางตรีโกณมิติของมุมที่ต้องการ
- **tan()** หาค่า Tangent ทางตรีโกณมิติของมุมที่ต้องการ
- **sqrt()** หาค่ารากที่สองของค่าคงที่หรือตัวแปร (Square Root)
- **pow()** หาค่ายกกำลังของค่าคงที่หรือตัวแปร
- **log()** หาค่า log ฐาน n (Natural Logarithm) ของค่าคงที่หรือตัวแปร
- **log10()** หาค่า log ฐาน 10 ของค่าคงที่หรือตัวแปร
- **max()** หาค่าสูงสุด

54

## Mathematical Functions (2)

- **min()** หาค่าต่ำสุด
- **bindec()** ใช้ในการเปลี่ยนเลขฐาน 2 ให้เป็นฐาน 10
- **octdec()** ใช้ในการเปลี่ยนเลขฐาน 8 ให้เป็นฐาน 10
- **hexdec()** ใช้ในการเปลี่ยนเลขฐาน 16 ให้เป็นฐาน 10
- **decbin()** ใช้ในการเปลี่ยนเลขฐาน 10 ให้เป็นฐาน 2
- **decoct()** ใช้ในการเปลี่ยนเลขฐาน 10 ให้เป็นฐาน 8
- **dechex()** ใช้ในการเปลี่ยนเลขฐาน 10 ให้เป็นฐาน 16
- **number\_format()** ใช้ในการกำหนดรูปแบบการแสดงผลของตัวเลข

55

## PHP Forms (1)

- ในการออกแบบฟอร์มรับข้อมูลผ่านทางหน้าเว็บเพจนั้น จะใช้ภาษา HTML เข้ามาเป็นตัวช่วยในการทำตรงจุดนี้ โดยฟอร์มที่ถูกสร้างขึ้นนี้จะทำการส่งค่าข้อมูลที่ผู้ใช้กรอกข้อมูล ไปประมวลผลยังไฟล์ php ที่อ้างอิงไว้ที่เครื่องบริการ หรือ Web Server โดยที่การเขียน HTML Form การอ้างอิงชื่อไฟล์ และการกำหนดวิธีการส่งข้อมูล มีรูปแบบดังนี้

```
<form method="วิธีการส่งข้อมูล" action="ชื่อไฟล์">
```

```
.....
```

```
.....แท็ก HTML Code ในการสร้างฟอร์ม
```

```
.....
```

```
</form>
```

56

## PHP Forms (2)

- การสร้างฟอร์มจะเริ่มการเปิดและปิดแท็ก <form>...</form> โดยค่าต่างๆ ที่อยู่ภายในแท็ก <form> เรียกว่า Attribute
- action ใช้ในการกำหนดชื่อไฟล์ php ที่เราจะส่งข้อมูลจากฟอร์มประมวลผล
- Method ใช้กำหนดรูปแบบในการส่งข้อมูลไปยัง Web Server แบบ GET กับ POST

GET เป็นการส่งข้อมูลให้กับ php ไปพร้อมกับ URL เช่น test.php?User=Kitisak โดยที่ชุดของสตริงต่อท้าย URL จะถูกเก็บไว้ที่ตัวแปร QUERY\_STRING

POST เป็นการส่งข้อมูลให้กับ php ไปที่ละบรรทัด

57

## PHP Forms (3)

- การสร้างช่องรับข้อมูล (Text Box)
- Textbox เป็นการแสดงช่องสำหรับให้ผู้ใช้กรอกข้อมูลใดๆ ไม่ว่าจะเป็นตัวอักษรหรือตัวเลข สามารถสร้าง Textbox โดยการเขียนโค้ด HTML ได้ดังนี้

รูปแบบ

```
<input type = "text" name = "ชื่ออ้างอิง" size = "ขนาดความยาว">
```

ข้อมูลที่ป้อนจะเห็นตามนั้น

58

## PHP Forms (4)

- การสร้างช่องรับรหัสผ่าน (Parssword Box)
- Parssword Box เป็นการแสดงช่องสำหรับให้ผู้ใช้กรอกข้อมูลใดๆ ไม่ว่าจะเป็นตัวอักษรหรือตัวเลข แต่เวลากรอกข้อมูลลงในช่องแสดงจะเป็น \* แทน สามารถสร้าง Parssword Box โดยการเขียนโค้ด HTML ได้ดังนี้

รูปแบบ

```
<input type = "Parssword" name = "ชื่ออ้างอิง" size = "ขนาดความยาว">
```

ข้อมูลที่ป้อนจะถูกแทนด้วยเครื่องหมาย \*

59

## PHP Forms (5)

- การสร้างช่องรับข้อมูลขนาดใหญ่ (Text Area)
- Text Area เป็นการแสดงช่องสำหรับให้ผู้ใช้กรอกข้อมูลใดๆ ไม่ว่าจะเป็นตัวอักษรหรือตัวเลขเช่นเดียวกับ Text Box แต่ Text Area จะสามารถกรอกข้อความได้เป็นจำนวนมาก สามารถสร้าง Text Area ได้โดยการเขียนโค้ด HTML ดังนี้

รูปแบบ

```
<textarea name = "ชื่ออ้างอิง" rows = "จำนวนแถว" cols="จำนวนคอลัมน์"></textarea>
```

เหมาะสำหรับการป้อนข้อมูลที่มีขนาดมากๆ เช่น รายละเอียด ความคิดเห็น

60

### PHP Forms (6)

- การสร้างปุ่มส่งข้อมูล (Submit Button)  
ปุ่ม **Submit** ใช้สำหรับกดเพื่อส่งข้อมูลในแบบฟอร์มไปยัง URL ที่ปรากฏในแอ็ททริบิวต์ **action** ของ **FORM** สามารถสร้างปุ่ม **Submit** ได้โดยการเขียนโค้ด **HTML** ดังนี้

รูปแบบ

```
<input type = "Submit" value = "ข้อความแสดงบนปุ่ม">
```

61

### PHP Forms (7)

- การสร้างยกเลิกข้อมูล (Reset Button)  
ปุ่ม **Reset** ใช้สำหรับคืนค่าของข้อมูลที่อยู่ทุก **element** ในฟอร์มให้เป็นค่าเริ่มต้น สามารถสร้างปุ่ม **Reset** ได้โดยการเขียนโค้ด **HTML** ดังนี้

รูปแบบ

```
<input type = "Reset" value = "ข้อความแสดงบนปุ่ม">
```

62

### PHP Forms (8)

- การสร้างตัวเลือกรายการ (List Box)  
**List Box** เป็นตัวเลือกที่อนุญาตให้ผู้ใช้เลือกเพียงรายการใดรายการหนึ่งจากหลายรายการเหล่านั้น โดยจะปรากฏเป็น **List** รายการให้เลือกเมื่อ **Click mouse** ที่แถบเมนูด้านข้าง

รูปแบบ กรณีเลือกได้รายการเดียว

```
<selete name = "ชื่ออ้างอิง">
    <input value = "ค่าที่ต้องการจัดเก็บ">ข้อความที่ต้องการแสดง</option>
</selete>
```

63

### PHP Forms (9)

รูปแบบ กรณีเลือกได้หลายรายการเดียว  
(โดยการกดปุ่ม <Ctrl> หรือ <Shift> ค้างไว้แล้ว Click mouse เลือก)

```
<select name = "ชื่ออ้างอิง[ ]" multiple>
    <option>ข้อความที่ต้องการแสดง
</select>
```

64



## PHP Forms (10)

- การสร้างรายการเลือกทางใดทางหนึ่ง (Radio Box)

**Radio Box** เป็นตัวเลือกที่ให้ผู้เลือกเพียงข้อใดข้อหนึ่งจากหลายรายการเหล่านั้น เมื่อเลือกรายการใดรายการหนึ่งแล้วจะไม่สามารถคลิกกรายการได้ แต่สามารถย้ายไปเลือกรายการอื่นในกลุ่มเดียวกันได้ สามารถสร้าง **Radio Box** ได้โดยการเขียนโค้ด **HTML** ดังนี้

รูปแบบ

```
<input type = "radio" name = "ชื่ออ้างอิง" value = "ค่าที่ต้องการจัดเก็บ">ข้อความแสดง
```

65

## PHP Forms (11)

- การสร้างรายการเลือกแบบ **Check Box**

**Check Box** เป็นการเลือกรายการจากกลุ่มรายการเช่นเดียวกับ **Radio Button** แต่ **Check Box** สามารถเลือกได้หลายรายการ สามารถสร้าง **Check Box** โดยการเขียนโค้ด **HTML** ดังนี้

รูปแบบ

```
<input type = "checkbox" name = "ชื่ออ้างอิง" value = "ค่าที่ต้องการจัดเก็บ">ข้อความแสดง
```

66

## Cookies (1)

- Cookies คือ ข้อมูลขนาดเล็กที่สรีปต์สามารถเก็บไว้บนเครื่องฝั่งไคลเอนท์ได้ ประโยชน์ของการใช้งาน Cookies คือใช้เก็บข้อมูลเกี่ยวกับผู้ใช้ โดยเมื่อผู้ใช้กลับมาใช้งานเว็บเพจหน้านั้นอีกครั้ง เว็บเพจนั้นจะสามารถระบุถึงผู้ใช้และจัดเตรียมเพจที่เหมาะสมกับการใช้งานให้อัตโนมัติ สามารถติดตั้ง Cookies บนเครื่องผู้ใช้โดยการส่ง HTTP เซคเตอร์ที่บรรจุข้อมูลที่อยู่ในรูปแบบดังนี้

```
Set - Cookies: NAME =VALUE;[expires = DATE;][path = PATH ;][domain = DOMAIN_NAME;][secure]
```

67

## Cookies (2)

การติดตั้ง Cookies จาก PHP

PHP มีคำสั่งในการเรียกใช้ Cookies ดังนี้

- ฟังก์ชัน `setcookie ( )`

เป็นฟังก์ชันสำหรับกำหนดค่าของ Cookies ค่าพารามิเตอร์ต่างๆ ในฟังก์ชัน `setcookie ( )` นั้น จะมีความหมายเดียวกันกับการ `Set -Cookies` ใน HTTP เซคเตอร์

รูปแบบ

```
Int setcookie (string name [,string value [, int expire [, string path [, string domain [, int secure]]]])
```

68

## ตัวอย่าง การใช้ฟังก์ชัน `setcookie ( )`

1. `<?>`
2. `setcookie ("ck", "Hello, My Cookie!",time()+3600, "/work/");`
3. `echo $HTTP_COOKIE_VARS["ck"];`
4. `?>`

69

## Cookies (3)

- ฟังก์ชัน `session_set_cookie_params ( )`  
ฟังก์ชันนี้จะทำการกำหนดค่าพารามิเตอร์ต่างๆ ให้ `cookie` ด้วย `session`

รูปแบบ

`void session_set_cookie_params (int lifetime [, string path [, string domain]])`

70

## Cookies (4)

- ฟังก์ชัน `session_get_cookie_params ( )`  
เป็นฟังก์ชันที่รับค่าพารามิเตอร์ต่างๆ จาก `cookie` ด้วย `session`  
โดยจะส่งค่ากลับมาเป็น `array` ที่เก็บค่า `lifetime`, `path` และ `domain`

รูปแบบ

`array session_get_cookie_params (void)`

71

## การเก็บ Session ID

- โดยทั่วไปแล้ว PHP จะใช้ `Session` เพื่อเรียกใช้งาน `cookie` อีกรหัส ซึ่ง `cookie` นี้สามารถเก็บค่า `session ID` ได้ด้วย ส่วนอีกวิธีหนึ่งที่ใช้ในการเก็บ `session ID` คือ การรวม `session ID` เข้ากับ `URL` วิธีนี้ `Session ID` จะถูกฝังเข้ากับ `link` และส่งไปพร้อมกัน โดยที่ `session ID` จะถูกเก็บเป็นตัวแปรค่าคงที่ชื่อ `SID` การส่งวิธีนี้สามารถเพิ่มเข้าไปในส่วนท้ายของ `link` ซึ่งจะคล้ายกับวิธีการ `GET` พารามิเตอร์ เช่น

`<A HREF = "link.php?<? = SID?>">`

72

## PHP Session

Session ใน PHP จะทำงานด้วย Session ID ซึ่งได้มาจากการสุ่มตัวเลข Session ID ถูกสร้างขึ้นโดย PHP และเก็บไว้บนเครื่องของผู้ใช้ตามอายุ Session ซึ่งสามารถส่งไปกับ URL ได้เลย Session id จะทำหน้าที่คล้ายกับคีย์ที่อนุญาตให้ใช้ได้กับตัวแปรเฉพาะเท่านั้น ตัวแปรดังกล่าว คือ ตัวแปร Session รายละเอียดของตัวแปรเหล่านี้ จะเก็บไว้ในเซิร์ฟเวอร์ ส่วน Session ID เป็นข้อมูลเดียวที่ปรากฏบนฝั่งไคลเอนต์ในช่วงเวลาที่ผู้ใช้ติดต่อไปยัง Session ID จะปรากฏผ่านทาง URL ซึ่งช่วยให้สามารถเข้าถึงตัวแปร Session ที่เก็บบนเซิร์ฟเวอร์ สำหรับ Session นั้นๆ ได้ โดยทั่วไปแล้วตัวแปร Session จะเก็บใน Flat file บนเซิร์ฟเวอร์

73

## การสร้างและการจัดการ Session

มี 4 ขั้นตอน ดังนี้

1. การเริ่มต้น Session
2. การกำหนดตัวแปร Session
3. การใช้ตัวแปร Session
4. การยกเลิกตัวแปร และการทำลาย Session

74

## การเริ่มต้น Session

ก่อนที่จะใช้ฟังก์ชันการทำงานของ Session นั้น จะต้องเรียกใช้ฟังก์ชันสำหรับเริ่มต้น Session ก่อน

การเริ่มต้น Session ด้วย `Session_start()`

ใช้สำหรับการเริ่มต้น session โดยจะตรวจสอบว่ามี Session ID อยู่แล้วหรือไม่ ถ้ามีอยู่แล้วจะโหลดตัวแปร Session ที่สามารถใช้งานได้รูปแบบ ดังนี้

```
session_start()
```

75

## การกำหนดตัวแปร Session

การกำหนดตัวแปร Session สามารถทำได้โดยใช้ฟังก์ชัน `Session_register()` ซึ่งจะทำให้สคริปต์อื่นๆสามารถใช้กับตัวแปร Session ตัวเดียวกันนี้ได้

ฟังก์ชัน `Session_register()` มีรูปแบบ ดังนี้

```
Session_register("name")
```

76

## การกำหนดค่าให้กับตัวแปร Session

หลังจากลงทะเบียนตัวแปร Session โดยใช้ฟังก์ชัน `session_register()` สามารถกำหนดค่าให้กับตัวแปรนั้นได้ โดยมีรูปแบบการใช้งาน ดังนี้

```
$_session["Session's name"] = "Session's value";
```

\*\* Session's name หมายถึง ชื่อของตัวแปร Session ที่จะกำหนด สามารถกำหนดได้หลายตัว

Session's value หมายถึง ค่าของ Session ที่ต้องการกำหนดให้กับตัวแปร

77

## ตัวอย่าง การกำหนดค่าให้กับตัวแปร Session

1. `<?php`
2. `session_start();`
3. `$_session ["count"]="50";`
4. `echo "คุณเข้าชมเว็บเพจหน้านี้เป็นจำนวน ".$_session["count"]." ครั้ง";`
5. `?>`

78

## การใช้ตัวแปร Session

การเรียกใช้ตัวแปร Session สามารถทำได้ 2 วิธี คือ

1. การเรียกใช้ตัวแปรแบบวิธีเดียวกับการเข้าถึงตัวแปรทั่วไป โดยวิธีนี้สามารถเข้าไปกำหนดในไฟล์ `php.ini` ที่ค่า `register_globals` ให้เป็น `on`
2. การเรียกใช้ตัวแปรผ่านทาง associative array `$HTTP_SESSION_VARS` เช่น

```
$HTTP_SESSION_VARS["var"];
```

ตัวแปร Session นี้ไม่สามารถถูกเขียนค่าทับด้วยข้อมูล GET หรือ POST ซึ่งนับเป็นคุณสมบัติที่ดีในการรักษาความปลอดภัย

79

## การตรวจสอบตัวแปร Session

เนื่องจากตัวแปรทั่วไปสามารถสร้างโดยผู้ใช้งานผ่านทางวิธี GET หรือ POST ได้ ดังนั้น จึงไม่สมควรที่จะตรวจสอบตัวแปร Session ด้วยฟังก์ชัน `isset()` หรือ `empty()` แต่ควรใช้ฟังก์ชัน `session_is_registered()` ในการตรวจสอบตัวแปร session แทน

ฟังก์ชันตรวจสอบตัวแปร

```
bool session_is_registered(string name)
```

โดยที่ `name` หมายถึง ชื่อตัวแปร session ที่ต้องการค้นหา

80

## ตัวอย่าง การตรวจสอบตัวแปร Session

- <?
- Session\_register ("count");
- if(session\_is\_registered("count")){
- echo "ตัวแปร session count ถูกสร้างแล้ว<br>";
- echo "คุณเข้าชมเว็บเพจหน้านี้เป็นจำนวน". \$count++.
- "ครั้ง";
- }
- else
- echo "ไม่พบตัวแปร session count";
- ?>

81

## การยกเลิกตัวแปร และการทำลายตัวแปร Session

ฟังก์ชัน Session\_unregister()

เมื่อสร้างตัวแปร Session แล้วหากไม่ต้องการตัวแปรนั้นอีก สามารถยกเลิกด้วยฟังก์ชัน

session\_unregister(name)

โดยที่ name หมายถึง ชื่อของตัวแปร Session

เมื่อต้องการจะปิด Session ควรจะทำการยกเลิกตัวแปร Session ทุกตัวก่อน แล้วจึงปิด Session ด้วยฟังก์ชัน Session\_destroy()

82

## การยกเลิกตัวแปร และการทำลายตัวแปร Session

ฟังก์ชัน Session\_destroy()

ฟังก์ชันสำหรับปิดการเชื่อมต่อ Session

Session\_destroy(void);

โดย void หมายถึง ชื่อของตัวแปร Session

83

## Class และ Object ของ PHP

Class หรือ Object เป็น โครงสร้างของโปรแกรมที่รวบรวมข้อมูล และฟังก์ชันไว้ด้วยกันอย่างเป็นหมวดหมู่ โดยในแต่ละ Class นั้นจะมี ข้อมูลและ Method สำหรับจัดการ โครงสร้างข้อมูลนั้นๆ

โครงสร้าง Class

การสร้าง Class คือการสร้าง โครงร่างของข้อมูลและฟังก์ชันที่เกี่ยวข้องเท่านั้นเมื่อต้องการ ใช้งาน ผู้ใช้จะต้องสร้างตัวแปรและกำหนดให้เป็นข้อมูลประเภท Class ที่กำหนด โครงสร้างของ Class จะประกอบด้วย ข้อมูล Method ต่างๆ และจะมี Method พิเศษอันหนึ่งซึ่งจะเป็น Construction ที่จะถูกเรียกเมื่อมีการสร้าง Class ทั้งนี้ฟังก์ชันที่เป็น Construction จะต้องมีชื่อเดียวกันกับ Class เสมอ

84

### ตัวอย่างการสร้าง Class

```

<?
class Queue {
    var $lists; // the list for queue
    var $max; // the max of queue
    var $curr; // the max of queue
    function Queue($maxlist = 100) {
        // Construction
        $max=$maxlist;
        $curr=0;
    }

    function enqueue($data) {
        // enqueue Method
        if ($curr==$maxlist)
            return -1; // Queue is full
        else {
            $lists[$curr++] = $data;
        }
    }

    function dequeue($data) {
        // dequeue Method
        if ($curr==0)
            return -1; // Queue is empty
        else {
            $data = $lists[$curr--];
        }
        return $data;
    }
}

$queue1 = new Queue(10);
$queue2 = new Queue();

$queue1->enqueue("data1");
$queue2->enqueue("data2");
?>

```

### การสร้าง Class ใหม่จาก Class เดิม

นอกจากนี้เรายังสามารถที่จะสร้าง class ใหม่ที่มีพื้นฐานมาจาก Class เดิมที่มีอยู่ได้อีกด้วย ซึ่งเราเรียกการทำงานในลักษณะนี้ว่า การ Derive Class ซึ่งการเรียกใช้จะเขียนในการทำงานเดียวกันกับการ Extend Class ของ Java ดังตัวอย่าง แสดงการสร้าง Queue แบบใหม่ที่สามารถจะดูจำนวนที่ว่างที่เหลือใน Queue ได้ ทั้งนี้การอ้างอิงถึง Class ที่กำลังทำงานด้วย เราจะใช้ตัวแปรชื่อ \$this เสมอ

### ตัวอย่าง

```

<?
class Ext_Queue extends Queue{
    function Ext_Queue($maxlist = 100) {
        // Construction
        $this->Queue($maxlist);
    }

    function RemainQueue() {
        // enqueue Method
        return $max-$curr;
    }
}

$queue1 = new Ext_Queue(10);

echo $queue1->RemainQueue();

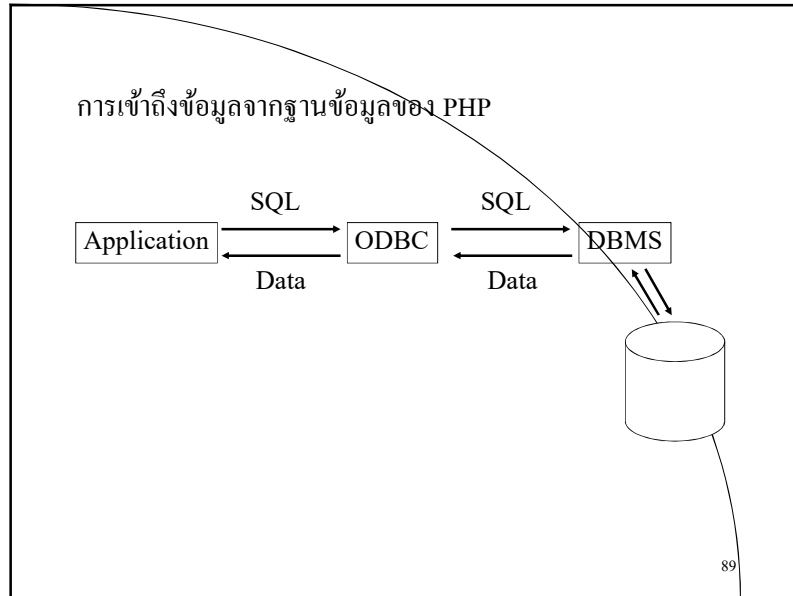
?>

```

\*\*ทั้งนี้ข้อควรระวังคือ Class ที่ Derived มาใหม่นี้จะไม่มีการเรียก Constructor ของ Class ที่เป็นต้นแบบมาด้วย

### การใช้ PHP ติดต่อกับฐานข้อมูล

- สร้างฐานข้อมูลโดยใช้ DBMS ต่างๆ เช่น Access และ SQL server
- กำหนด Data Source Name ใน ODBC ที่จะใช้ติดต่อกับฐานข้อมูลนั้น
- ใช้ ADO Data Component ในการติดต่อกับ Data Source Name



การใช้ PHP ติดต่อกับฐานข้อมูล

การใช้ PHP ติดต่อกับฐานข้อมูล มี 2 ส่วน

1. โดยส่วนแรกใช้ SQL server ซึ่ง SQL server จะสนับสนุนผู้ใช้ UNIX และ Windows
2. จะเป็นการติดต่อโดยใช้ ODBC ซึ่งการใช้ ODBC จะจำกัดอยู่เฉพาะผู้ใช้งาน Windows เท่านั้น

90

คำสั่งภาษา SQL ที่จำเป็นต้องใช้(1)

การเพิ่มข้อมูลเข้าสู่ตาราง

รูปแบบ

```
INSERT INTO<ชื่อตาราง>(field1,field2)VALUES(data1,data2)
```

การเรียกดูข้อมูลจากตาราง

รูปแบบ

```
SELECT<field>
FROM<ชื่อ>
```

91

คำสั่งภาษา SQL ที่จำเป็นต้องใช้(2)

การปรับข้อมูลในตาราง

รูปแบบ

```
UPDATE<ชื่อตาราง>
SET<field>=<ข้อมูล>
WHERE<เงื่อนไข>
```

92

### คำสั่งภาษา SQL ที่จำเป็นต้องใช้(3)

#### การเปิดการติดต่อกับฐานข้อมูล MySQL

การสร้าง Connection ในการเปิดการติดต่อกับฐานข้อมูล MySQL เราจะใช้ฟังก์ชัน mysql-connect() โดยต้องระบุค่าอักขระ 3 ตัว คือ “ชื่อ Host”, “ชื่อผู้ใช้”, “รหัสผ่าน”

93

### คำสั่งภาษา SQL ที่จำเป็นต้องใช้(4)

#### การเลือกฐานข้อมูลมาใช้งาน (Select Database)

หลังจากสร้าง Connection ได้แล้ว เราต้องเลือกฐานข้อมูลที่จะทำงานด้วย โดยการเรียกใช้งานฟังก์ชัน mysql-select-db() ซึ่งต้องระบุอักขระ 2 ตัว คือ “ชื่อฐานข้อมูล”, “Connection ที่ได้สร้างไว้”

94

### คำสั่งภาษา SQL ที่จำเป็นต้องใช้(5)

#### การปิดการติดต่อกับฐานข้อมูล MySQL

การยกเลิกหรือการปิด Connection ที่ติดต่อกับฐานข้อมูล MySQL เราจะใช้ฟังก์ชัน mysql-close() โดยระบุค่าอักขระ 1 ตัว คือ “Connection” ที่ต้องการติดต่อ

95

### ติดต่อฐานข้อมูล Access ด้วย ODBC

การที่เราจะเขียน php ติดต่อกับฐานข้อมูล Microsoft Access เราจะต้องติดต่อผ่าน ODBC ซึ่งเป็น Driver ของการเชื่อมต่อฐานข้อมูล ซึ่งการติดต่อผ่าน ODBC จะต้องมีการ Set ชื่อ DSN เพื่อเป็นชื่ออ้างอิงในการติดต่อกับฐานข้อมูลนั้น ๆ

96



### ฟังก์ชันที่ใช้ติดต่อฐานข้อมูล Access ด้วย ODBC (1)

ฟังก์ชัน	ความหมาย
odbc_close()	ปิดการเชื่อมต่อกับฐานข้อมูล
odbc_close_all()	ปิดการเชื่อม ODBC ทั้งหมด
odbc_commit()	สั่งทำคำสั่งกับ ODBC
odbc_connect()	เชื่อมต่อกับฐานข้อมูล หรือ datasource
odbc_do()	เตรียมและเอ็กคิวทีวส์คำสั่ง SQL
odbc_exec()	เตรียมและเอ็กคิวทีวส์คำสั่ง SQL
odbc_execute()	เอ็กคิวทีวส์คำสั่งที่เตรียมไว้แล้ว

97

### ฟังก์ชันที่ใช้ติดต่อฐานข้อมูล Access ด้วย ODBC(2)

ฟังก์ชัน	ความหมาย
odbc_fetch_into()	ดึงข้อมูลครั้งละ 1 Record และเก็บข้อมูลลง Array
odbc_fetch_row()	ดึงข้อมูลครั้งละ 1 Record
odbc_field_name()	แสดงชื่อฟิลด์
odbc_field_type()	แสดงประเภทของฟิลด์
odbc_field_len()	แสดงค่าขนาดของฟิลด์
odbc_num_files()	นับจำนวนฟิลด์
odbc_pconnect()	เปิดการเชื่อมฐานข้อมูลที่มีอยู่

98

### ฟังก์ชันที่ใช้ติดต่อฐานข้อมูล Access ด้วย ODBC(3)

ฟังก์ชัน	ความหมาย
odbc_perpare()	เตรียมคำสั่งสำหรับการเอ็กคิวทีวส์
odbc_num_rows()	หาจำนวน Record ของข้อมูล
odbc_result()	แสดงข้อมูลออกมา
odbc_result_all()	แสดงข้อมูลเป็นตารางของ Html
odbc_rollback()	ทำคำสั่งอีกครั้ง
odbc_setoption()	ปรับแต่งการตั้งค่า Odbc จะให้ค่า false ถ้าเกิดข้อผิดพลาดนอกจากเป็นจริง

99