

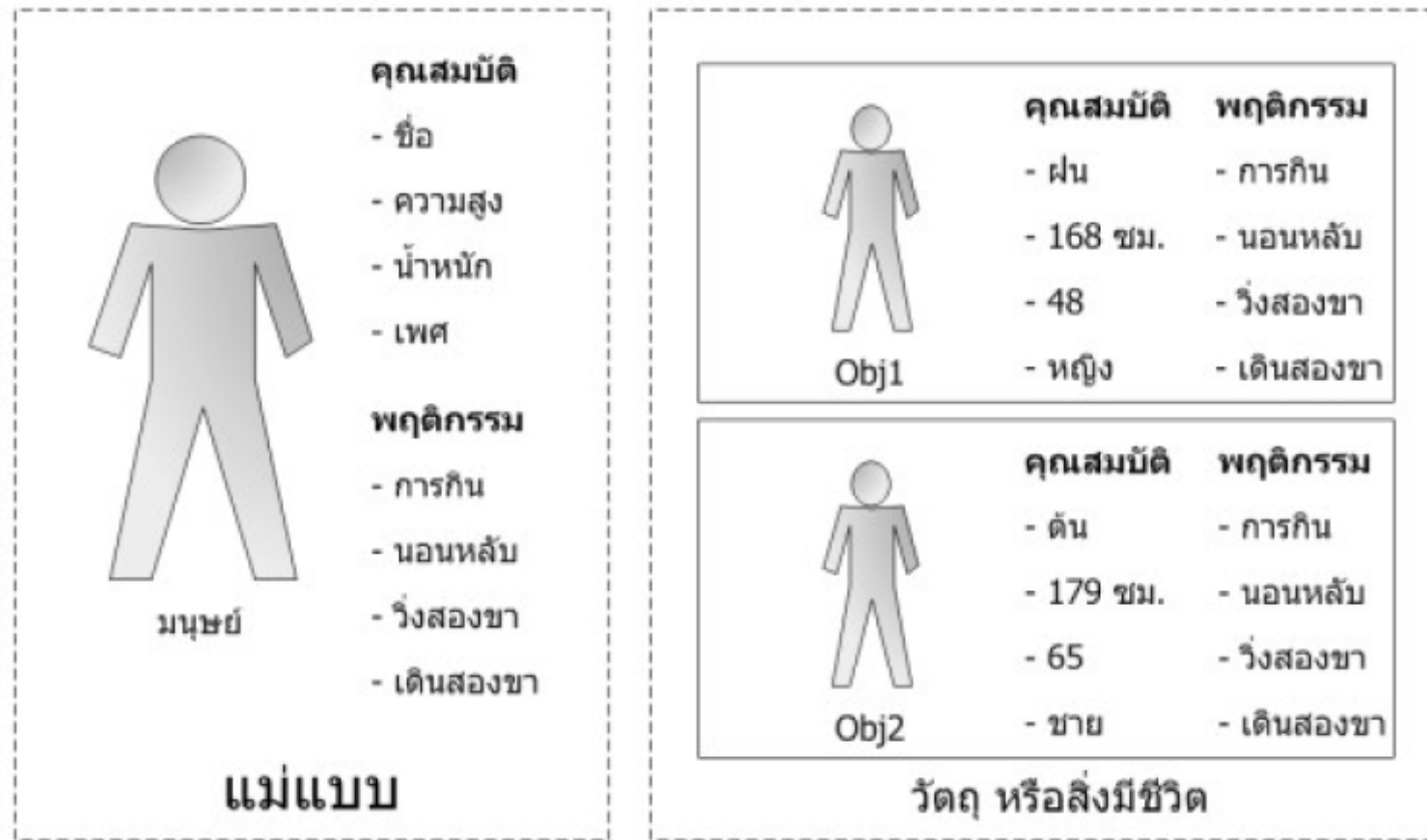
# PHP OOP + PHP Mailer

# การเขียนแบบเชิงวัตถุ (Object Oriented Programming)

Class ประกอบด้วย

- คุณสมบัติ (Properties)
- แบบแผนพฤติกรรม (Method Behavior หรือ Method)

# การเขียนแบบเชิงวัตถุ (Object Oriented Programming)



ภาพแสดงแม่แบบ และวัตถุ

## การเขียนแบบเชิงวัตถุ (Object Oriented Programming)

- แม่แบบ Class
- ลูกแบบ (Sub-Class)
- วัตถุ (Object)
- การส่งข้อความ (Messaging)
- ข้อความ (Message)
- การห่อหุ้มข้อมูล (Encapsulation)
- การสืบทอด (Inheritance)
- การทำหลายรูปแบบ (Polymorphism)
- คุณสมบัติ (Properties)
- แบบแผนพฤติกรรม (Method Behavior)

## การเขียนโปรแกรมเชิงวัตถุใน PHP

คำที่ใช้เป็นคำหลัก (Keyword) ในการเขียนได้แก่

- class
- function
- this
- parent
- private, protected, public

# การเขียนโปรแกรมเชิงวัตถุใน PHP

## รูปแบบคำสั่ง

```
<?php  
class classname {  
  
}  
?>
```

รูปแบบคำสั่งที่ 1

1. คำหลักที่เราจะใช้นั้นคือ **class**
2. หลังคำหลัก **class** ตามด้วยชื่อแม่แบบ (**classname**)
3. เปิดและปิดด้วยเครื่องหมายปีกกา เพื่อกำหนดช่วงการเขียนชุดคำสั่งของแม่แบบ

# การเขียนโปรแกรมเชิงวัตถุใน PHP

## รูปแบบคำสั่ง (Constructor Method)

```
1. <?php
2. class class_name {
3.     function class_name([parameter]){
4.         ....
5.         [return ....;]
6.     }
7. }
8. ?>
```

## การเขียนโปรแกรมเชิงวัตถุใน PHP

```
<?php
class BaseClassPHP4 {/** PHP 4 */
    function BaseClassPHP4() {print "In BaseClass constructor\n";}
}
class SubClassPHP4 extends BaseClassPHP4 {
    function SubClassPHP4() {
        $this->BaseClassPHP4();
        print "In SubClass constructor\n";
    }
}
class BaseClassPHP5 {/** PHP 5 */
    function __construct() {print "In BaseClass constructor\n";}
}
class SubClassPHP5 extends BaseClassPHP5 {
    function __construct() {
        parent::__construct();
        print "In SubClass constructor\n";
    }
}
$obj = new BaseClassPHP4();
$obj = new SubClassPHP4();
$obj = new BaseClassPHP5();
$obj = new SubClassPHP5();
?>
```



# การเขียนโปรแกรมเชิงวัตถุใน PHP

## ตัวอย่างชุดคำสั่ง

```
1. <?php
2. class human {
3.     var $name = "human";// default name = "human"
4.     var $Sex= 1; // default = 1 and 1 = male, 2 = female
5.     var $height = 0; // height > 0
6.     var $weight = 0; // weight > 0
7.     var $age = 0; // age > 0
8.     function human($name, $sex){
9.         $this->setName($name);
10.        $this->setSex($sex);
11.        $this->setHeight(170);
12.        $this->setWeight(60);
13.        $this->setAge(21);
14.    }
```

# การเขียนโปรแกรมเชิงวัตถุใน PHP

## รูปแบบคำสั่ง (Inheritance)

```
<?php
class human {
    function Talk() {
        echo "Hello World";
    }
}
class male extends human {
    . . . .
}
class female extends human {
    . . . .
}
?>
```

## การเขียนโปรแกรมเชิงวัตถุใน PHP

ตัวอย่างชุดคำสั่ง

```
<?php
class human {
    function Talk() {
        echo "Hello World";
    }
}
?>
```

## ตัวอย่าง

```
<?php
class human {
    var $name = "human";// default name = "human"
    var $Sex= 1;          // default = 1 and 1 = male, 2 = female
    var $height = 0; // height > 0
    var $weight = 0; // weight > 0
    var $age      = 0; // age > 0
    function setName($value = 'human'){
        $this->name = $value;
    }
    function setSex($value = 1){
        $this->Sex = ($value == 1 || $value == 2 ? $value : 1);
    }
    function setHeight($value = 1){
        $this->height = ($value > 0 ? $value : 1 );
    }
    function setWeight($value = 1){
        $this->weight = ($value > 0 ? $value : 1 );
    }
    function setAge($value = 1){
        $this->age = ($value > 0 ? $value : 1 );
    }
    function getName(){
        return $this->name;
    }
    function getSex(){
        return ($this->Sex == 1 ? 'male' : 'female');
    }
    function getHeight(){
        return $this->height;
    }
    function getWeight(){
        return ($this->Sex == 2 ? 'No Answer' : $this->weight);
    }
    function getAge(){
        return ($this->Sex == 2 ? 'No Answer' : $this->age);
    }
}
?>
```

## การเขียนโปรแกรมเชิงวัตถุใน PHP

รูปแบบการเรียกใช้คำสั่ง

```
$objectname = new classname();
```

ตัวอย่างชุดคำสั่ง

```
<?php
class human {
    function Talk() {
        echo "Hello World";
    }
}
$somsak = new human();
$somsak->Talk();
?>
```

## การเขียนโปรแกรมเชิงวัตถุใน PHP

การกำหนดการเข้าถึงคุณสมบัติและแบบแผนพฤติกรรม

- public
- protected
- private

# การเขียนโปรแกรมเชิงวัตถุใน PHP

ตัวอย่าง

```
1. <?php
2. /** Define MyClass */
3. class MyClass {
4.     public $public = 'Public';
5.     protected $protected = 'Protected';
6.     private $private = 'Private';
7.     function printHello() {
8.         echo $this->public;
9.         echo $this->protected;
10.        echo $this->private;
11.    }
12. }
13. /** Define MyClass2 */
14. class MyClass2 extends MyClass {
15.     protected $protected2 = 'Protected2';
16.     function printHello() {
17.         echo $this->public;
18.         echo $this->protected;
19.         echo $this->protected2;
20.         echo $this->private;
21.     }
22. }
```

## การเขียนโปรแกรมเชิงวัตถุใน PHP

### ตัวอย่าง

```
1. $obj = new MyClass();
2. echo $obj->public; // works
3. echo $obj->protected; // Fatal Error
4. echo $obj->private; // Fatal Error
5. $obj->printHello(); // Shows Public, Protected and Private
6.
7. $obj2 = new MyClass2();
8. echo $obj2->public; // works
9. echo $obj2->private; // Undefined
10. echo $obj2->protected; // Fatal Error
11. $obj2->printHello(); // Shows Public, Protected, Protected2, not Private
12. ?>
```



## ตัวอย่าง

```
1. <?php
2. /** Define MyClass */
3. class MyClass{
4.     // Constructors must be public
5.     public function __construct() { }
6.     // Declare a public method
7.     public function MyPublic() { }
8.     // Declare a protected method
9.     protected function MyProtected() { }
10.    // Declare a private method
11.    private function MyPrivate() { }
12.    // This is public
13.    function Foo(){
14.        $this->MyPublic();
15.        $this->MyProtected();
16.        $this->MyPrivate();
17.    }
18.}
19.$myclass = new MyClass;
20.$myclass->MyPublic(); // Works
21.$myclass->MyProtected(); // Fatal Error
22.$myclass->MyPrivate(); // Fatal Error
23.$myclass->Foo(); // Public, Protected and Private work
24./** Define MyClass2 */
25.class MyClass2 extends MyClass {
26.    // This is public
27.    function Foo2(){
28.        $this->MyPublic();
29.        $this->MyProtected();
30.        $this->MyPrivate(); // Fatal Error
31.    }
32.}
33.$myclass2 = new MyClass2;
34.$myclass2->MyPublic(); // Works
35.$myclass2->Foo2(); // Public and Protected work, not Private
36.??>
```

# PHP Mailer

## Features

- Can send emails with multiple TOs, CCs, BCCs and REPLY-TOs
- Redundant SMTP servers
- Multipart/alternative emails for mail clients that do not read HTML email
- Support for 8bit, base64, binary, and quoted-printable encoding
- Uses the same methods as the very popular AspEmail active server (COM) component
- SMTP authentication
- Word wrap
- Address reset functions
- HTML email
- Tested on multiple SMTP servers: Sendmail, qmail, Postfix, Imail, Exchange, Mercury, Courier
- Works on any win32 or \*nix platform
- Flexible debugging
- Custom mail headers
- Multiple fs, string, and binary attachments (those from database, string, etc)
- Embedded image support

# PHP Mailer

## Property

- \$From
- \$FromName
- \$Sender
- \$Subject
- \$Body
- \$AltBody
- \$Host
- \$Port
- \$Hostname
- \$SMTPAuth
- \$Username
- \$Password

## Method

- Send()
- AddAddress(\$address, \$name = "")
- AddCC(\$address, \$name = "")
- AddBCC(\$address, \$name = "")
- AddReplyTo(\$address, \$name = "")
- AddAttachment(\$path, \$name = "", \$encoding = "base64", \$type = "application/octet-stream")
- IsSMTP()
- MsgHTML(\$message)
- IsMail()
- IsSendmail()

<http://phpmailer.worxware.com/index.php?pg=properties>

<http://phpmailer.worxware.com/index.php?pg=methods>

# PHP Mailer

ตัวอย่าง

```
require("class.phpmailer.php");

$mail = new PHPMailer();

$mail->IsSMTP(); // send via SMTP
$mail->Host = "smtp.domain.com"; // SMTP servers
$mail->SMTPAuth = true; // turn on SMTP authentication
$mail->Username = "email@domain.com"; // SMTP username
$mail->Password = "password"; // SMTP password

$mail->From = "email@domain.com";
$mail->FromName = "Name";
$mail->AddAddress("Recipient@emailaddress.com", "Name");
$mail->AddReplyTo("yourname@domain.com", "Your Name");

$mail->WordWrap = 50; // set word wrap

$mail->IsHTML(true); // send as HTML

$mail->Subject = "Here is the subject";
$mail->Body = "This is the HTML body";
$mail->AltBody = "This is the text-only body";

if(!$mail->Send())
{
    echo "Message was not sent";
    echo "Mailer Error: " . $mail->ErrorInfo;
    exit;
}

echo "Message has been sent";
```

## เอกสารอ้างอิง – PHP Mailer

- <http://www.askapache.com/php/phpfreaks-eric-rosebrocks-phpmailer-tutorial.html>
- <http://orangsetya.wordpress.com/2007/09/12/send-mail-using-smtp-authentication-phpmailer-script/>
- <http://phpmailer.worxware.com/index.php?pg=tutorial>