

PHP's Object Oriented capabilities

Fajiang Yu, fjyu@whu.edu.cn

School of Computer, Wuhan University

2017.4



Agenda

- 1 class
- 2 constructor
- 3 private
- 4 extends



Agenda

- 1 class
- 2 constructor
- 3 private
- 4 extends



class

```
<?php
// PHP 5 class definition
class Bear {
    // define properties
    public $name;
    public $weight;
    public $age;
    public $sex;
    public $colour;
```



```
// define methods
public function eat() {
    echo $this->name." is eating...";
}
public function run() {
    echo $this->name." is running...";
}
public function kill() {
    echo $this->name." is killing prey...";
}
public function sleep() {
    echo $this->name." is sleeping...";
}
}
?>
```



class

```
<?php
// my first bear
$daddy = new Bear;
// give him a name
$daddy->name = "Daddy Bear";
// how old is he
$daddy->age = 8;
// what sex is he
$daddy->sex = "male";
// what colour is his coat
$daddy->colour = "black";
// how much does he weigh
$daddy->weight = 300;
```



```
// give daddy a wife
$mommy = new Bear;
$mommy->name = "Mommy Bear";
$mommy->age = 7;
$mommy->sex = "female";
$mommy->colour = "black";
$mommy->weight = 310;

// and a baby to complete the family
$baby = new Bear;
$baby->name = "Baby Bear";
$baby->age = 1;
$baby->sex = "male";
$baby->colour = "black";
$baby->weight = 180;
```



```
// a nice evening in the Bear family
// daddy kills prey and brings it home
$daddy->kill();
// mommy eats it
$mommy->eat();
// and so does baby
$baby->eat();
// mommy sleeps
$mommy->sleep();
// and so does daddy
$daddy->sleep();
// baby eats some more
$baby->eat();

?>
```



class

```
<?php
// PHP 4 class definition
class Bear {
    // define properties
    var $name;
    var $weight;
    var $age;
    var $sex;
    var $colour;
```



```
// define methods
function eat() {
    echo $this->name." is eating...";
}
function run() {
    echo $this->name." is running...";
}
function kill() {
    echo $this->name." is killing prey...";
}
function sleep() {
    echo $this->name." is sleeping...";
}
}
?>
```



Agenda

- 1 class
- 2 constructor**
- 3 private
- 4 extends



constructor

```
<?php
// PHP 5 class definition
class Bear {
    // define properties
    public $name;
    public $weight;
    public $age;
    public $colour;

    // constructor
    public function __construct() {
        $this->age = 0;
        $this->weight = 100;
        $this->colour = "brown";
    }
    // define methods
}
?>
```



constructor

```
<?php
// PHP 4 class definition
class Bear {
    // define properties
    var $name;
    var $weight;
    var $age;
    var $colour;

    // constructor
    function Bear() {
        $this->age = 0;
        $this->weight = 100;
        $this->colour = "brown";
    }
    // define methods
}
?>
```



```
<?php
    // create instance
    $baby = new Bear;
    $baby->name = "Baby Bear";
    echo $baby->name." is ".$baby->colour."
        and weighs ".$baby->weight." units at birth";
?>
```



Agenda

- 1 class
- 2 constructor
- 3 private
- 4 extends



private

PHP 5 makes it possible to mark class properties and methods as **private**, which means that they cannot be manipulated or viewed outside the class definition. This is useful to protect the inner workings of your class from manipulation by object instances.

```
<?php
// PHP 5 class definition
class Bear {
    // define properties
    public $name;
    public $age;
    public $weight;
    private $_lastUnitsConsumed;
    // constructor
    public function __construct() {
        $this->age = 0;
        $this->weight = 100;
        $this->_lastUnitsConsumed = 0;
    }
}
```




```
// define methods
public function eat($units) {
    echo $this->name." is eating ".$units.
        " units of food...";
    $this->weight += $units;
    $this->_lastUnitsConsumed = $units;
}
public function getLastMeal() {
    echo "Units consumed in last meal were ".
        $this->_lastUnitsConsumed." ";
}
}
?>
```



```
<?php
    $bob = new Bear;
    $bob->name = "Bobby Bear";
    $bob->eat(100);
    $bob->eat(200);
    echo $bob->getLastMeal();

    // the next line will generate a fatal error
    $bob->_lastUnitsConsumed = 1000;

?>
```



Agenda

- 1 class
- 2 constructor
- 3 private
- 4 extends



extends

create a new class based on an existing class, add new features

```
<?php
// PHP 5 class definition
class Bear {
    // define properties
    public $name;
    public $weight;
    public $age;
    public $sex;
    public $colour;

    // constructor
    public function __construct() {
        $this->age = 0;
        $this->weight = 100;
    }
}
```



```
// define methods
public function eat($units) {
    echo $this->name." is eating ".
        $units." units of food...";
    $this->weight += $units;
}
public function run() {
    echo $this->name." is running...";
}
public function kill() {
    echo $this->name." is killing prey...";
}
public function sleep() {
    echo $this->name." is sleeping...";
}
}
```



```
// extended class definition
class PolarBear extends Bear {
    // constructor
    public function __construct() {
        parent::__construct();
        $this->colour = "white";
        $this->weight = 600;
    }

    // define methods
    public function swim() {
        echo $this->name." is swimming...";
    }
}

?>
```



```
<?php
// create instance of Bear()
$tom = new Bear;
$tom->name = "Tommy Bear";

// create instance of PolarBear()
$bob = new PolarBear;
$bob->name = "Bobby Bear";
// $bob can use all the methods of Bear() and PolarBear()
$bob->run();
$bob->kill();
$bob->swim();

// $tom can use all the methods of Bear() but not PolarBear()
$tom->run();
$tom->kill();
$tom->swim();

?>
```



```
<?php
// PHP 4 extended class definition
class PolarBear extends Bear {
    // constructor
    function PolarBear() {
        parent::Bear();
        $this->colour = "white";
        $this->weight = 600;
    }
    // define methods
    function swim() {
        echo $this->name." is swimming...";
    }
}
?>
```



To prevent a class or its methods from being inherited, use the **final** keyword before the class or method name (this is new in PHP 5 and will not work in older versions of PHP).

```
<?php
// PHP 5 class definition
final class Bear {
    // define properties
    // define methods
}
```



```
// extended class definition
// this will fail because Bear() cannot be extended
class PolarBear extends Bear {
    // define methods
}

// create instance of PolarBear()
// this will fail because Bear() could not be extended
$bob = new PolarBear;
$bob->name = "Bobby Bear";
echo $bob->weight;

?>
```



Thank You!
Any Questions?

