

## PHP

- 1 PHP简介
- 2 PHP基本语法
- 3 PHP流程控制
- 4 PHP函数
- 5 PHP字符串操作
- 6 PHP数组
- 7 PHP时间和日期
- 8 PHP面向对象
  - 8.1 PHP面向对象简介
  - 8.2 PHP class：定义类
  - 8.3 PHP实例化对象
  - 8.4 PHP构造函数
  - 8.5 PHP析构函数
  - 8.6 PHP继承
  - 8.7 PHP \$this
  - 8.8 PHP命名空间
  - 8.9 PHP设计模式
  - 8.10 PHP魔术方法**
  - 8.11 PHP抽象类和抽象方法
  - 8.12 PHP interface：接口
  - 8.13 PHP最终类和最终方法
  - 8.14 PHP clone关键字

[首页](#) > [PHP](#) > [PHP面向对象](#)

阅读：3,837

## PHP魔术方法（附带实例）



小白入职大厂完全攻略，很硬很肝  
学习路线 / 笔试面试 / 升职加薪 / 跳槽技巧

[猛击查看详情](#)

C语言中文网推出辅导班啦，包括「C语言辅导班、C++辅导班、算法/数据结构辅导班」，全部都是**一对一教学**：一对一辅导 + 一对一答疑 + 布置作业 + 项目实践 + 永久学习。QQ在线，随时响应！

在面向对象编程中，**PHP** 提供了一系列的魔术方法，这些魔术方法为编程提供了很多便利，在 PHP 中的作用是非常重要的。PHP 中的魔术方法通常以\_\_（两个下划线）开始，并且不需要显式的调用而是在某种特定条件下自动调用的。

前面我们介绍的构造函数 \_\_construct 和析构函数 \_\_destruct 就属于 PHP 中的魔术方法。PHP 中的魔术方法如下表所示：

魔术方法	作用
__construct()	实例化类时自动调用
__destruct()	类对象使用结束时自动调用
__set()	在给未定义的属性赋值时自动调用
__get()	调用未定义的属性时自动调用
__isset()	使用 isset() 或 empty() 函数时自动调用
__unset()	使用 unset() 时自动调用

- 8.15 PHP判断对象是否属于某个类
- 8.16 PHP自动加载机制
  - 9 正则表达式
- 10 PHP会话控制
- 11 PHP错误和异常处理
- 12 MySQL数据库的基础操作
- 13 PHP文件目录操作
- 14 PHP图像处理

PHP魔术方法（附带实例）

__sleep()	使用 serialize 序列化时自动调用
__wakeup()	使用 unserialize 反序列化时自动调用
__call()	调用一个不存在的方法时自动调用
__callStatic()	调用一个不存在的静态方法时自动调用
__toString()	把对象转换成字符串时自动调用
__invoke()	当尝试把对象当方法调用时自动调用
__set_state()	当使用 var_export() 函数时自动调用，接受一个数组参数
__clone()	当使用 clone 复制一个对象时自动调用
__debugInfo()	使用 var_dump() 打印对象信息时自动调用

前面我们已经介绍过了《[构造函数](#)》和《[析构函数](#)》，下面我们再来介绍几个常用的魔术方法。

## 1、\_\_set() 方法

在为当前环境下未定义或不可见的类属性赋值时，会自动调用 \_\_set() 方法。定义该方法的语法格式如下：

```
public function __set($key, $value){
    ... .. ;
}
```

其中，参数 \$key 是要操作的变量名称，\$value 为变量 \$key 的值。

【示例】下面使用简单的示例来演示一下 \_\_set() 方法的使用。

```
01.  <?php
02.      class Website{
03.          public $name;
04.          private $url;
```



```
05.         public function __set($key, $value){
06.             echo '为“'.$key.'”赋值“'.$value.'”失败！<br>';
07.         }
08.     }
09.
10.     $object      = new Website();
11.     $object -> name = 'C语言中文网';
12.     $object -> url  = 'http://c.biancheng.net/php/';
13.     $object -> title = 'PHP教程';
14.     ?>
```

运行结果如下：

为“url”赋值“http://c.biancheng.net/php/”失败！  
为“title”赋值“PHP教程”失败！

利用 \_\_set() 方法的特性，我们可以通过 \_\_set() 方法为类中的使用 private 关键词修饰的属性进行赋值或修改。如下所示：

```
01. <?php
02.     class Website{
03.         public $name;
04.         private $url = '';
05.         public function __set($key, $value){
06.             if(isset($this->$key)){
07.                 $this -> $key = $value;
08.             }else{
09.                 echo '为“'.$key.'”赋值“'.$value.'”失败！<br>';
10.             }
11.         }
12.         public function getUrl(){
13.             echo $this -> url;
14.         }
```

```
15.     }
16.
17.     $object          = new Website();
18.     $object -> name   = 'C语言中文网';
19.     $object -> url    = 'http://c.biancheng.net/php/';
20.     $object -> title = 'PHP教程';
21.     $object -> getUrl();
22.  ?>
```

运行结果如下：

为“title”赋值“PHP教程”失败！  
http://c.biancheng.net/php/

## 2、\_\_get() 方法

在调用或获取当前环境下未定义或不可见的类属性时，会自动调用 \_\_get() 方法，定义该方法的语法格式如下：

```
public function __get($name){
    ... ..;
}
```

参数 \$name 为要操作的变量名称。

【示例】下面使用简单的示例来演示一下 \_\_get() 方法的使用。

```
01.  <?php
02.     class Website{
03.         public $url = 'http://c.biancheng.net/php/';
04.         private $name = 'C语言中文网';
05.         public function __get($name){
06.             echo '获取：“'.$name.'”失败！’;
```



```
07.         }  
08.     }  
09.  
10.     $object = new Website();  
11.     echo $object -> url.'<br>';  
12.     echo $object -> name.'<br>';  
13.     echo $object -> title.'<br>';  
14.     ?>
```

运行结果如下：

```
http://c.biancheng.net/php/  
获取： "name" 失败!  
获取： "title" 失败!
```

### 3、\_\_isset() 方法

当在类外部对类中不可访问或不存在的属性使用 `isset()` 或 `empty()` 函数时，会自动调用 `__isset()` 方法，该方法的语法格式如下：

```
public function __isset($name){  
    ... ..;  
}
```

参数 `$name` 为要访问的属性名称。

`isset()` 函数可以检查一个变量是否存在并且不为 `NULL`，传入一个变量作为参数，如果传入的变量存在则传回 `true`，否则传回 `false`。

`empty()` 函数可以检查一个变量是否为空，同样需要传入一个变量作为参数，如果变量并不存在，或者变量的值等于 `FALSE`，那么这个变量会被认为不存在。



通过前面的学习我们知道，类中的公有成员可以在类外访问，而私有成员则无法在类外访问。也就是说，我们可以使用 `isset()` 或 `empty()` 函数来检查类中的公有属性是否存在，而对类中的私有属性这两个函数就无效了。

如果想要使用 `isset()` 或 `empty()` 函数对类中的私有属性进行检测的话，我们只需要在类中添加一个 `__isset()` 方法就可以了，当在类外部使用 `isset()` 或 `empty()` 函数时，会自动调用类里面的 `__isset()` 方法。

【示例】下面使用简单的示例来演示一下 `__isset()` 方法的使用。

```
01. <?php
02.     class Website{
03.         public $url = 'http://c.biancheng.net/php/';
04.         private $name = 'C语言中文网';
05.         public function __isset($name){
06.             if(property_exists('Website', $name)){
07.                 echo '成员属性: “'.$name.'” 存在! <br>';
08.             }else{
09.                 echo '成员属性: “'.$name.'” 不存在! <br>';
10.             }
11.         }
12.     }
13.
14.     $object = new Website();
15.     isset($object -> title);
16.     isset($object -> name);
17. ?>
```

运行结果如下：

成员属性: “title” 不存在!  
成员属性: “name” 存在!

提示：property\_exists() 函数可以用来检测类中是否定义了该属性，语法格式为 property\_exists(\$class\_name, \$property\_name)，其中 \$class\_name 为字符串形式的类名，即判断类 class\_name 中是否定义了 property\_name 属性。

## 4、\_\_unset() 方法

当在类外部对类中不可访问或不存在的属性使用 unset() 函数时，\_\_unset() 方法会被自动调用，该方法的语法格式如下：

```
public function __unset($name){  
    ... ..;  
}
```

参数 \$name 为要访问的属性名称。

我们先来看一下 unset() 函数，unset() 函数的作用是删除指定的变量，需要传入一个或多个变量作为参数，另外，该函数没有返回值。

同样，我们也可以使用 unset() 函数在类外部去删除类中的成员属性。与上面介绍的 \_\_isset() 方法相似，如果要删除类中的公有属性的话直接使用 unset() 函数即可；如果要删除类中的私有属性的话，则需要在类中添加一个 \_\_unset() 方法。

【示例】下面使用简单的示例来演示一下 \_\_unset() 方法的使用。

```
01.  <?php  
02.      class Website{  
03.          public $url = 'http://c.biancheng.net/php/';  
04.          private $name = 'C语言中文网';  
05.          public function __unset($name){  
06.              if(property_exists('Website', $name)){  
07.                  unset($this->$name);
```



```
08.             echo '移除成员属性: “’. $name.’” 成功! <br>’;  
09.             }else{  
10.             echo '成员属性: “’. $name.’” 不存在! <br>’;  
11.             }  
12.         }  
13.     }  
14.  
15.     $object = new Website();  
16.     unset($object->url);  
17.     unset($object->name);  
18.     unset($object->abc);  
19.     ?>
```

运行结果如下:

移除成员属性: “name” 成功!  
成员属性: “abc” 不存在!

## 5、\_\_call() 方法

当调用类中一个不可访问或不存在的方法时，\_\_call() 方法会被调用。该方法的语法格式如下:

```
public function __call($name, $arguments){  
    ... .. ;  
}
```

其中，\$name 为要调用的方法名称，\$arguments 为传递给 \$name 的参数所组成的数组。

当调用的方法不存在时会自动调用 \_\_call() 方法，程序会继续执行下去，从而可以避免当调用方法不存在时产生错误所导致的程序终止。

【示例】下面使用简单的示例来演示一下 \_\_call() 方法的使用。



```
01. <?php
02.     class Website{
03.         public function say() {
04.             echo 'Welcome C语言中文网! <br>';
05.         }
06.         public function __call($name, $arguments) {
07.             echo '你所调用的方法: '.$name;
08.             if(!empty($arguments)){
09.                 echo ' 【以及参数: ';
10.                 print_r($arguments);
11.                 echo ' 】';
12.             }
13.             echo ' 不存在! <br>';
14.         }
15.     }
16.
17.     $obj = new Website();
18.     $obj -> say();
19.     $obj -> url('http://c.biancheng.net/php/');
20.     $obj -> title();
21. ?>
```

运行结果如下:

```
Welcome C语言中文网!
你所调用的方法: url 【以及参数: Array ( [0] => http://c.biancheng.net/php/ ) 】 不存在!
你所调用的方法: title 不存在!
```

## 6、\_\_clone() 方法



可以使用 clone 关键字复制对象，当复制完成时，如果定义了 \_\_clone() 方法，则新创建的对象（复制生成的对象）中的 \_\_clone() 方法会被自动调用，通过该方法我们可以做一些必要的操作。\_\_clone() 方法的语法格式如下：

```
public function __clone(){  
    ... ..;  
}
```

该函数不需要传入参数。

【示例】下面使用简单的示例来演示一下 \_\_clone() 方法的使用。

```
01. <?php  
02.     class Website{  
03.         public function say() {  
04.             echo 'Welcome C语言中文网! <br>';  
05.         }  
06.         public function __clone() {  
07.             echo '对象克隆成功<br>';  
08.         }  
09.     }  
10.  
11.     $obj = new Website();  
12.     $obj2 = clone $obj;  
13.     echo '<pre>';  
14.     var_dump($obj, $obj2);  
15. ?>
```

运行结果如下：

```
对象克隆成功  
object(Website)#1 (0) {  
}
```



```
object(Website)#2 (0) {  
}
```

关注微信公众号「站长严长生」，在手机上阅读所有教程，随时随地都能学习。本公众号由[C语言中文网站](#)长运营，每日更新，坚持原创，敢说真话，凡事有态度。



微信扫描二维码关注公众号

### 优秀文章

进制转换：二进制、八进制、十六进制、十进制之

Android应用程序发布的步骤

C语言2048小游戏源码下载和思路解析

JS switch case语句

动辄一两万的编程培训班是不是骗人的？能不能参

数据结构和算法的关系和区别

JSP Request.getSession()方法：获取和请求相关

AJAX跨域请求（案例实战）

JSON数据格式和变量声明

fn.startsWith()和fn.endsWith()函数

精美而实用的网站，分享优质编程教程，帮助有志青年。千锤百炼，只为大作；精益求精，处处斟酌；这种教程，看一眼就倾心。

[关于网站](#) | [关于站长](#) | [如何完成一部教程](#) | [联系我们](#) | [网站地图](#)

Copyright ©2012-2022 biancheng.net, 陕ICP备15000209号

*biancheng.net*

