

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: 4,153

PHP extends: 继承



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!

面向对象编程 (OOP) 的一大好处就是, 可以使用一个类继承另一个已有的类, 被继承的类称为父类或基类, 而继承这个父类的类称为子类。子类可以继承父类的方法和属性, 因此通过继承可以提高代码的重用性, 也可以提高软件的开发效率。

子类可以增加父类之外的新功能, 因此也可以将子类称为父类的“扩展”。此外, 子类还可以继承父类的构造函数, 当子类被实例化时, PHP 会先在子类中查找构造函数。如果子类有自己的构造函数, PHP 会先调用子类中的构造函数。当子类中没有时, PHP 则会去调用父类中的构造函数。

在 PHP 中, 类的继承需要通过 extends 关键字来实现。语法格式如下所示:

```
class 子类名 extends 父类名{  
    ...  
}
```

在 C++ 中, 一个子类可以继承一个基类, 也可以继承多个基类。继承一个基类称为单继承; 继承多个基类称为多继承。但在 PHP 中没有多继承, 只能使用单继承模式。也就是说, 一个类只能直接从另一个类中继承数据。但一个类可以有多个子类。

- 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

通过前面的学习我们知道，类中的成员属性和方法都需要使用访问权限修饰符来修饰，这是 PHP 面向对象编程中的一个重要特性。不同访问修饰符的作用及其之间的区别如下表所示：

	public (默认)	private	protected
同一个类中访问	√	√	√
在子类中访问	√		√
在类的外部访问	√		

1、继承 public 成员

类的成员都有不同的访问权限，由访问控制关键字来控制的。在 PHP 中，父类所有使用 public 修饰的成员均可以被子类继承。

【示例】定义一个基类，使用另一类来继承这个基类，同时在尝试使用子类调用基类中的成员方法。

```
01.  <?php
02.      class Website{
03.          public $name, $url, $title;
04.          public function __construct() {
05.              echo '-----基类中的构造函数-----<br>';
06.          }
07.          public function demo() {
08.              echo '基类中的成员方法<br>';
09.          }
10.      }
11.
12.      class ClassOne extends Website{
13.
14.      }
15.
```



```
16.     class ClassTwo extends Website{
17.         public function __construct() {
18.             echo '-----子类中的构造函数-----<br>';
19.         }
20.     }
21.
22.     $object = new ClassOne();
23.     $object -> demo();
24.     $object2 = new ClassTwo();
25.     $object2 -> demo();
26.     ?>
```

运行结果如下:

```
-----基类中的构造函数-----
基类中的成员方法
-----子类中的构造函数-----
基类中的成员方法
```

2、继承 protected 成员

很多情况下有些类继承的属性是不想在类外部被访问的,这时就可以把这个成员声明为一个保护成员,也就使用 protected 修饰这个成员。受保护的成员不可以在类外部访问到,但是可以在子类的内部访问,也就是说我们可以在子类设置一个成员函数来访问这个受保护成员。

【示例】将上面示例的代码略作调整,演示类中 protected 成员的继承情况。

```
01.  <?php
02.     class Website{
03.         public $name, $url, $title;
04.         public function __construct() {
05.             echo '-----基类中的构造函数-----<br>';
```

```
06.         }
07.         protected function demo() {
08.             echo '基类中的成员方法<br>';
09.         }
10.     }
11.
12.     class ClassOne extends Website{
13.
14.     }
15.
16.     class ClassTwo extends Website{
17.         public function __construct() {
18.             echo '-----子类中的构造函数-----<br>';
19.         }
20.         public function test() {
21.             $this->demo();
22.         }
23.     }
24.
25.     $object = new ClassOne();
26.     // $object -> demo();      // 在子类中调用父类使用 protected 修饰的成员方法会报错
27.     $object2 = new ClassTwo();
28.     $object2 -> test();
29.     ?>
```

运行结果如下:

```
-----基类中的构造函数-----
-----子类中的构造函数-----
基类中的成员方法
```



在子类中调用父类使用 `protected` 修饰的成员方法会报错，如果去掉上面代码中第 26 行的注释，程序会报错并停止运行，如下所示：

```
Fatal error: Uncaught Error: Call to protected method Website::demo() from context " ... ..
```

3、private 成员

类中使用 `private` 修饰的成员被称为私有成员。父类中的私有成员不会被子类继承，因此不能被子类访问到，下面来看一个示例。

```
01. <?php
02.     class Website{
03.         private function demo(){
04.             echo '基类中的成员方法<br>';
05.         }
06.     }
07.
08.     class ClassOne extends Website{
09.         public function test(){
10.             $this->demo();
11.         }
12.     }
13.
14.     $object = new ClassOne();
15.     $object->test();
16. ?>
```

运行结果如下：

```
Fatal error: Uncaught Error: Call to private method Website::demo() from context 'ClassOne' ... ..
```

在子类中调用父类中使用 `private` 关键字修饰的成员时，程序会报错并终止运行。



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

快速排序算法，C语言快速排序算法详解

Shell编程系列视频教程（赶星老师出品10集）

Hibernate hibernate.default_catalog属性：指定

VS Code运行Python程序

面向连接和无连接的套接字到底有什么区别

使用netwox构造IP数据包

Python基础/Python进阶一对一答疑

IoT是什么意思？

Maven插件（plugin）

JS generator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

