

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：9,328

PHP class：定义类



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！

PHP 支持面向对象编程，支持类和对象的概念。从数据类型的角度看，对象是一种比较特殊的数据类型。它由一个事先定义好的类生成，而类由用户自己定义，它由一系列数据和对这些数据操作的数个函数组成。

初步使用面向对象程序设计，要掌握的就是使用面向对象的思想抽象出一个类，这是一种思想的转化，难点也就在这里。而这些思想体现在程序中就是代码。在这里，要写出代码、掌握语法，都是很简单，而能正确地抽象出一个类才重点。

下面先从简单的语法开始逐渐深入地讲解类。

1、定义类

在 PHP 中，可以使用 class 关键字加类名的方式定义一个类，然后用大括号 {} 将在类体中定义类的属性和方法包裹起来，类的语法格式如下：

```
[修饰类的关键字] class 类名{  
    类的属性和方法;  
}
```

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

类名和变量名还有函数名的命名规则相似，都需要遵循 PHP 中的自定义命名规则，可以是任何非 PHP 保留字的合法标签。一个合法类名以字母或下划线开头，后面跟着若干字母，数字或下划线。如果类名由多个单词组成，习惯上将每个单词的首字母大写。另外，类名最好具有一定的意义，不要随便由几个字母组成。

修饰类的关键字是一个可选参数，可以省略。我们通常使用下面这些关键字来修饰类：

- **abstract**：抽象类或方法，被修饰为抽象类之后，类将不能被实例化，但可以被继承。如果类中有至少一个方法被声明为抽象的，那么这个类也必须声明为抽象的。继承一个抽象类的时候，子类必须重新定义父类中的所有抽象方法，而且这些方法的访问控制必须和父类中一样。
- **final**：使用 **final** 修饰的类不能被继承，而使用 **final** 修饰的方法不能在子类中重新定义。

注意：一个类可以包含有属于自己的常量、变量（在类中称为“成员属性”或者“属性”）以及函数（在类中称为“成员方法”或者“方法”）。

2、成员属性

在类中直接声明的变量称为成员属性（也可以称为成员变量），可以在类中声明多个变量，即对象中可以有多个成员属性，每个变量都存储对象不同的属性信息。语法格式如下：

```
访问权限修饰符 属性名称 = 属性值;
```

成员属性的类型可以是 PHP 中的标量类型和复合类型，所以也可以是其他类实例化的对象，但在类中使用资源和空类型是没有意义的。

前面我们绍过，声明变量时不需要任何关键字修饰，但是在类中声明成员属性时，变量前面一定要使用一个关键字来修饰，例如 **public**、**private**、**static** 等，但这些关键字修饰的变量都具有一定的意义。如果不需要有特定意义的修饰，可以使用“**var**”关键字，一旦成员属性有其他的关键字修饰就需要去掉“**var**”。

常用访问权限修饰符及其含义如下所示：

- **public**：公共的，在类的内部、子类中或者类的外部都可以使用，不受限制；
- **protected**：受保护的，在类的内部和子类中可以使用，但不能在类的外部使用；
- **private**：私有的，只能在类的内部使用，在类的外部或子类中都无法使用。

注意：一个类，即一对大括号之间的全部内容都要在一段代码中，也就是一个 `<?php ... ?>` 之间，不能分割成多块。

【示例】创建一个 `Students` 类并在类中声明一些成员属性，代码如下所示：

```
01. <?php
02.     class Students{
03.         var $name;
04.         public $age;
05.         private $sex;
06.         public static $school;
07.     }
08. ?>
```

提示：权限修饰符可以和定义静态变量的关键字 `static` 混合在一起使用，如上面代码中所示。

3、成员方法

在类中定义的函数被称为成员方法。函数和成员方法唯一的区别就是，函数实现的是某个独立的功能，而成员方法是实现类中的一个行为，是类的一部分。

可以在类中声明多个成员方法，成员方法的声明和函数的声明完全一样，只不过在声明成员方法时可以在 `function` 关键字的前面加一些访问权限修饰符来控制访问权限，例如 `public`、`private`、`protected` 等。

另外需要注意的是，声明的成员方法必须和类是相关的，不能是一些没有意义的操作。例如，在声明学生类时，如果声明了“飞行”的成员方法，实例化出来的每个学生都可以飞了，这样就是一个设计上的错误。

【示例】在上面示例中创建的 `Students` 类中创建一些成员方法。

```
01. <?php
02.     class Students{
```

```
03.         var $name;
04.         public $age;
05.         private $sex;
06.         public static $school;
07.         public function Write() {
08.
09.         }
10.         protected static function Read() {
11.
12.         }
13.         function Listen() {
14.
15.         }
16.     }
17.     ?>
```

成员方法前面的权限修饰符可以省略，省略后默认权限为 public。在类中成员属性和成员方法的声明都是可选的，可以同时存在，也可以单独存在，具体可以根据实际情况而定。

在 PHP7 中，引入了类型声明，我们可以为成员方法的形参和返回值声明类型，格式如下所示：

```
[权限修饰符] function 方法名 (类型 参数1, 类型 参数2, ..., 类型 参数n) : 返回值类型 {
    ... ..
}
```

PHP7 中支持声明的参数类型包括整型、浮点型、字符串型和布尔类型。示例代码如下所示：

```
01. <?php
02.     class Students{
03.         var $name;
04.         public $age;
05.         private $sex;
06.         public static $school;
```

```
07.         public function Write(string $a, int $b):bool{
08.
09.     }
10.         protected static function Read(string $str):int{
11.
12.     }
13.         function Listen(int $num):bool{
14.
15.     }
16.     }
17.     ?>
```

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

Go语言map元素的删除和清空

Linux Apache安装过程详解（LAMP环境搭建）

C++ stable_sort(STL stable_sort)排序算法详解

MyBatis自定义TypeHandler处理枚举

JSP Request.getAttributeNames()方法：返回所 JSP

SpringCloud组件:搭建Eureka服务注册中心 JS实现动画效果（利用定时器）

Java Map集合详解 Pandas Panel三维数据结构

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