

## 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: 6,793

# PHP interface: 接口



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!

继承的特性简化了对象、类的创建, 增加了代码的重用性。但 PHP 只支持单继承, 也就是说每个类只能继承一个父类。为了解决这个问题, PHP 引入了接口。接口是一种特殊的抽象类, 而抽象类又是一种特殊的类, 所以可以将接口看作是一种特殊的类。

## 1、接口的声明

如果抽象类中的所有方法都是抽象方法, 我们就可以使用另外一种声明方式——“接口”技术。我们都知道类的声明是使用“class”关键字, 而接口的声明则是使用“interface”关键字。声明接口的格式如下所示:

```
interface 接口名称{  
    // 常量成员  
    // 抽象方法  
}
```

接口中所有的方法都是抽象方法, 而且不需要在方法前使用 abstract 关键字进行修饰。而且在接口中也不需要显示地使用 public 访问权限来进行修饰, 因为默认权限就是 public 的, 也只能是 public (公有的)。另外, 接口中不能声明变量, 只能使用 const 关键字声明为常量类型的成员属性。

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

接口和抽象类一样也不能实例化为对象，它是一种更严格的规范，也需要通过子类来实现。与抽象类不同的是，接口可以直接使用接口名称在接口外面获取常量成员的值。示例代码如下所示：

```
01.  <?php
02.      interface Demo{
03.          const NAME = 'C语言中文网';
04.          const URL = 'http://c.biancheng.net/php/';
05.          function fun1();
06.          function fun2();
07.      }
08.
09.      echo Demo::NAME.'<br>';
10.      echo Demo::URL;
11.  ?>
```

运行结果如下：

```
C语言中文网
http://c.biancheng.net/php/
```

## 2、接口的应用

因为接口不能进行实例化操作，所以要使用接口中的成员，就必须借助子类。在子类中继承接口需要使用 `implements` 关键字，如果实现多个接口的继承，那么每个接口之间使用逗号 `,` 分隔。

在使用 `implements` 关键字继承接口的同时，还可以使用 `extends` 关键字来继承一个类。也就是说，可以在继承一个类的同时实现多个接口，但一定要先使用 `extends` 继承类再去使用 `implements` 实现多个接口。语法格式如下：

```
class 类名 extends 父类名 implements 接口一, 接口二, ..., 接口 n {
    // 实现所有接口中的抽象方法
}
```



下面通过一个简单的示例来演示一下接口的使用，代码如下：

```
01. <?php
02.     interface Demo{
03.         const NAME = 'C语言中文网';
04.         const URL = 'http://c.biancheng.net/php/';
05.         function fun1();
06.         function fun2();
07.     }
08.
09.     class Website implements Demo{
10.         public function fun1(){
11.             echo self::NAME.'<br>';
12.         }
13.         public function fun2(){
14.             echo self::URL;
15.         }
16.     }
17.
18.     $obj = new Website();
19.     $obj -> fun1();
20.     $obj -> fun2();
21. ?>
```

运行结果如下：

```
C语言中文网
http://c.biancheng.net/php/
```

提示：既然要通过子类继承了接口中的方法，那么接口中的所有方法都必须在子类中实现，否则 PHP 将抛出如下所示的错误信息：



Fatal error: Class Website contains 1 abstract method and must therefore be declared abstract or implement the remaining methods (Demo::fun2) ... ..

我们还可以使用 `extends` 关键字让一个接口去继承另一个接口，实现接口之间的扩展。在下面的示例中先定义一个接口 `One`，然后再定义一个接口 `Two` 来继承 `One`：

```
interface One{
    function fun1();
    function fun2();
}
interface Two extends One{
    function fun3();
    function fun4();
}
```

如果需要使用抽象类去实现接口中的部分方法，也需要使用 `implements` 关键字。示例代码如下所示：

```
interface Demo{
    function fun1();
    function fun2();
}
abstract class Website implements Demo{
    public function fun2(){
        ... ..;
    }
    abstract function fun3();
}
```

说了这么多，那么使用接口具体有什么作用呢？其实它的作用很简单，当有很多人一起开发一个项目时，每个人都可能会去调用别人写的一些类。这时有人就会问了，我怎么知道别人的某个功能的实现方法是怎么命名的呢？这个时候 PHP 接口就起到作用了。



简单来说，我们可以将接口看作一个类的模板或者类的规定。如果你属于这类，你就必须遵循这个类的规定，少一个都不行，但是具体怎么去做，那是你的事。也就是说我们可以定义一些接口，每个接口中都包含若干的抽象方法。在多人开发时，每个人都根据自己的需要来实现一部分接口，这样就可以避免我们在调用别人开发的方法时不知道方法名的尴尬了。

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

Java Object类详解：Object类的常用方法

Linux启动引导程序（GRUB）加载内核的过程

MySQL删除存储过程（DROP PROCEDURE）

C# Directory类：文件夹操作

HBase过滤器入门教程

Java面向对象：对象的概念及面向对象的三个基本

Qt多窗口编程详解

C#正则表达式（Regex类）

JSP Request.getRequesteSessionId()方法：获

绘制E-R图：数据库概要设计阶段

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](#)



*biancheng.net*

