# 简介

自 PHP 5.4.0 起，PHP 实现了一种代码复用的方法，称为 trait。

Trait 是为类似 PHP 的单继承语言而准备的一种代码复用机制。

基本上就是Trait定义一组公用方法、属性，其它有需要的类可以直接拿去用，起到一个与class组合的作用。

Trait的定义与class差不多，只是不能实例化。

# 特性

特性如下：

* trait 中可以定义属性、方法、静态属性、静态方法，语法与class相同
* trait 中可以定义抽象方法, 语法与抽象类相同，需在调用的class实现
* 可以进行方法权限的控制
* Class可以引入多个Trait，如有重名方法需要在class里面指定用哪个Trait的方法，或者重命名某个Trait的同名方法
* 引入方法与class现有方法重名时, 其优先级顺序为: 当前类方法>trait方法>父类方法
* 在trait 代码块中还可以引入其他trait代码块, 语法与类中引入相同

# 方法同名冲突解决



# 实例



# 总结

## 抽象类、接口有什么区别?

抽象类与接口一样，都可以定义一组公共的方法, 一种规范。

但抽象类可以将子类中公共的方法进行抽离, 提高代码的复用。

抽象类可以实现多个接口，所以接口也可以定义抽象类的规范。

## trait与抽象类有什么区别?

trait 与抽象类看着很像, 功能相似，但侧重不同。

trait 侧重于定义共用的代码片段, 它并不管引用此片段的类是干什么的, 只要你需要, 就可以拿走去用, 简单说就是 trait 并不约束类的类型, 而仅仅定义方法。

抽象类则不同, 既然你继承我, 那你干的事情肯定跟我是一样的, 不能抽象类是去与数据库交互的, 你子类去写脚本了. 所以抽象类相对 trait 更侧重于 类型的约束。