

# 接口

## 1. 简述

类：有属性，有方法，有类常量

抽象类：有属性，有方法，有类常量，有抽象方法

更抽象类：接口：有接口常量、有抽象方法

可见，接口中只有两类简单的特性信息

接口常量：即常量

抽象方法：只有方法头——下级类自然应该具体实现

## 2. 接口作用

可看作抽象类更高层面的抽象规范，不考虑接口常量的话，相当规定了下级类要做什么——至于怎么做，不要求。

这和抽象类很像，但接口不通过继承，且可以实现类似其他语言的多继承效果。

Eg飞接口，鸟抽象类，鸵鸟不能飞(不用实现飞接口)，但鸵鸟是鸟(继承抽象类)

## 3. 注：接口默认都抽象，下级类必须实现该抽象方法(写出方法实现) 否则也需声明为抽象方法，待下一级类去实现

一个类只能继承一个父类，但可以实现多个接口，语法形式：

`class 类A extends 类 B implements 接口1, 接口2, ... {};`

一个接口同样可以继承另一个接口(单继承)

`interface 接口1 extends 接口1 {接口定义};`

与抽象类的区别：

(1) interface的抽象方法都需要被实现，抽象类不需要子类全部实现

(2) 一个类可以多实现接口，但，不能多继承抽象类

(3) interface强调功能，abstract强调所属关系

```
interface AAA{
    const PI=2.55;
    function attact();
}

class A implements AAA{
    public $proArr = array();//定义存储不存在的属性

    function attact(){
        echo "A类攻击";
        echo self::$PI;
    }
}

$a = new A();
$a->attact();
```