

重载技术

1. 简述

通常面向对象重载技术指的是：在一个类中，有多个同名方法，每个方法参数不同，顺序不同或者类型不同，即称为重载。

但，php不能定义多个方法名一致的函数——语法错误

实际上，php的重载技术指：

- (1) 属性重载，当使用同一个不存在的属性，就自动调用预先定义好的某个方法__get(\$param)、__set(\$param, \$value) 以处理该属性；
- (2) 方法重载，当使用一个不存在的方法，就自动调用类预先定义好的方法__call(\$method_name, \$param_arr){}、__staticcall(\$method_name, \$param_arr){}；以处理该行为。

2. 属性重载

- (1) 取值:\$v1 = 对象-> 属性; //自动调用类中的__get() 函数
- (2) 赋值:对象-> 属性=; //自动调用类中的__set() 函数
- (3) 判断是否存在:isset(\$对象->属性); //自动调用类中__isset() 函数
- (4) 销毁对象:unset(\$对象->属性); //自动调用类中__unset() 函数

注：前提都是，类中预先写好这些方法实现

可以实现未定义的属性，却可以使用该些属性

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

    //重载属性
    function __get($name){
        if(isset($this->proArr[$name])){
            return $this->proArr[$name];
        }
        return "属性未定义";
    }

    function __set($name, $value){
        $this->proArr[$name] = $value;
    }

    function __isset($name){
        if(isset($this->proArr[$name])){
            return true;
        }
        return false;
    }

    function __unset($name){
        unset($this->proArr[$name]);
    }
}
```

3. 方法重载

可以让其他对象处理该方法调用或调用其他方法，利用这点，可以实现其他语言传统意义上的重载

```

function f1(){
    echo "掉了内部f1";
}

//重载方法
//比如重载f1(), fa($a);
function __call($name, $arguments){
    if (count($arguments) == 0){
        $this->f1();
    }else if(count($arguments) == 1){
        f1($arguments[0]);
    }
}

}

$a = new A();
$a->abc("d");

function f1($a){
    echo "掉了外部f1";
}

```

静态方法的重载方法为

`__staticcall() {$name, $arguments};`