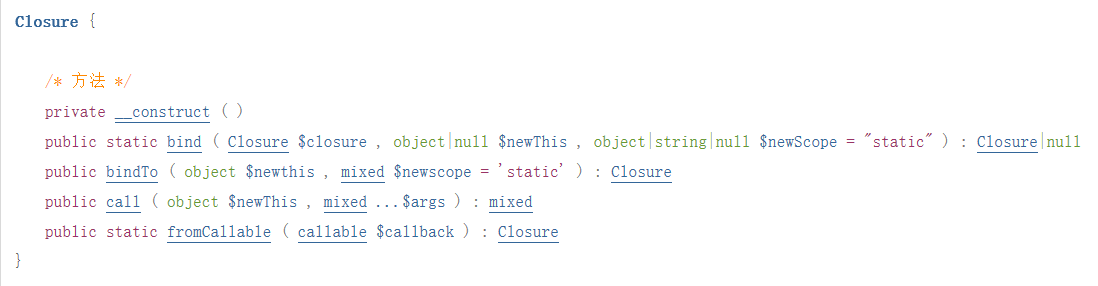
# 简介

用于代表 匿名函数的类.  
  
匿名函数会产生这种类型的对象。这个类带有一些方法允许在匿名函数创建后对其进行更多的控制。  
除了此处列出的方法，还有一个 \_\_invoke 方法。这是为了与其他实现了 \_\_invoke()魔术方法的对象保持一致性，但调用匿名函数的过程与它无关。

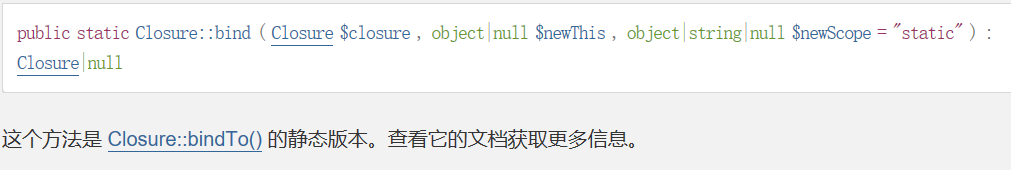
# 类摘要

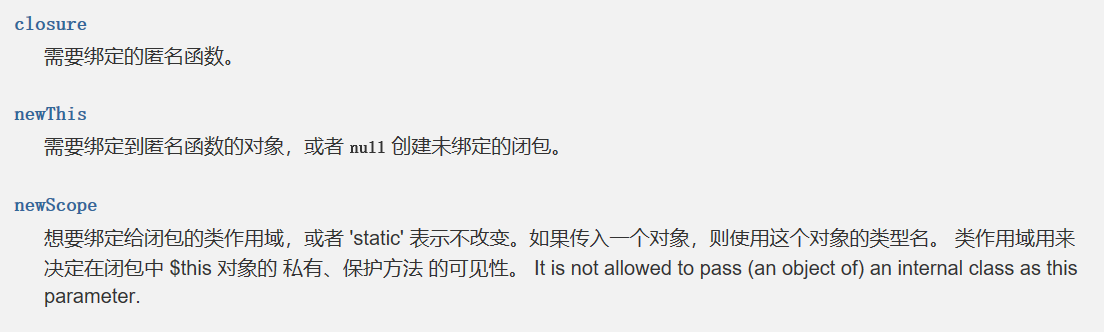


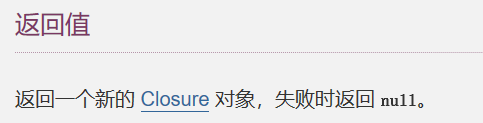
方法作用  
•Closure::\_\_construct — 用于禁止实例化的构造函数  
•Closure::bind — 复制一个闭包，绑定指定的$this对象和类作用域。  
•Closure::bindTo — 复制当前闭包对象，绑定指定的$this对象和类作用域。  
•Closure::call — 绑定并调用闭包  
•Closure::fromCallable — 将 callable 转换为闭包

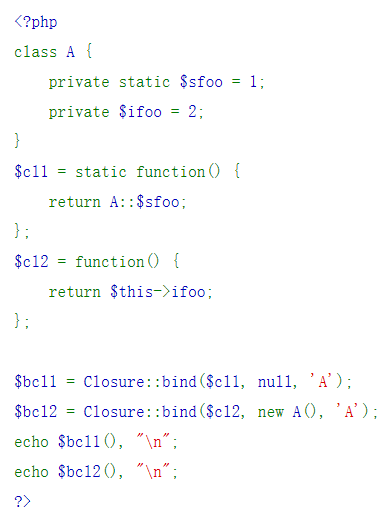
## Closure::bind

Closure::bind — 复制一个闭包，绑定指定的$this对象和类作用域。







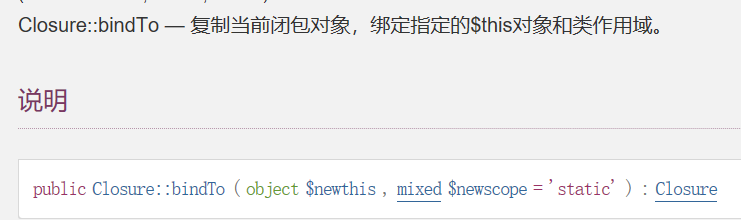


以上例程的输出类似于：

1

2

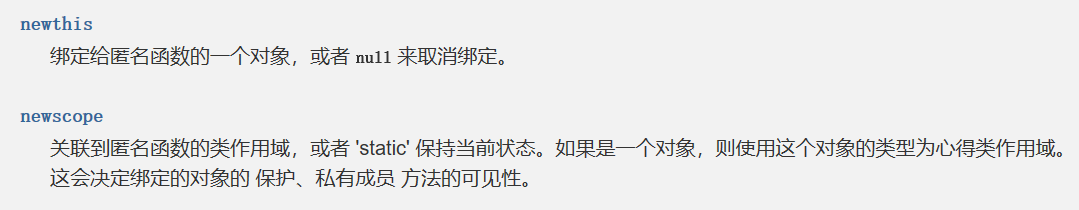
## Closure::bindTo

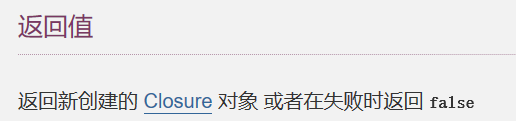


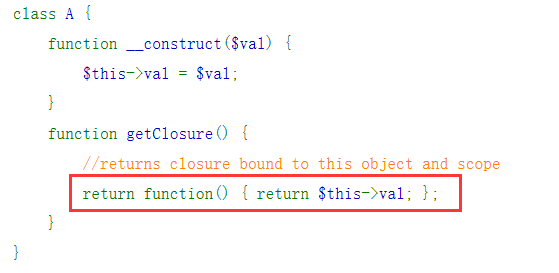
创建并返回一个 [匿名函数](mk:@MSITStore:C:\\Users\\张卫航\\Desktop\\php_enhanced_zh.chm::/res/functions.anonymous.html)， 它与当前对象的函数体相同、绑定了同样变量，但可以绑定不同的对象，也可以绑定新的类作用域。

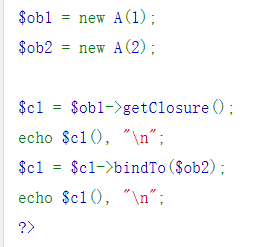
"绑定的对象"决定了函数体中的 $this 的取值，"类作用域"代表一个类型、决定在这个匿名函数中能够调用哪些 私有 和 保护 的方法。 也就是说，此时 $this 可以调用的方法，与 newscope 类的成员函数是相同的。

静态闭包不能有绑定的对象（ newthis 参数的值应该设为 null）不过仍然可以用 bubdTo 方法来改变它们的类作用域。









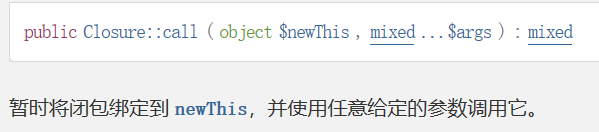
以上例程的输出类似于：

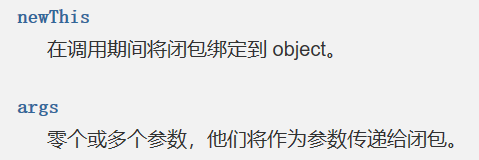
1

2

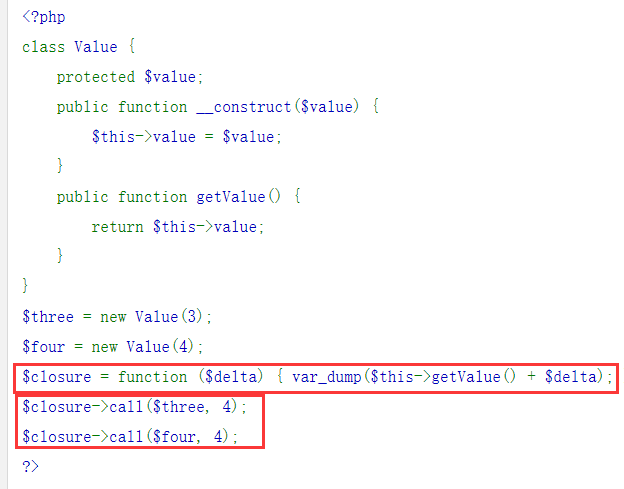
## Closure::call

Closure::call — 绑定并调用闭包





返回闭包的返回值。



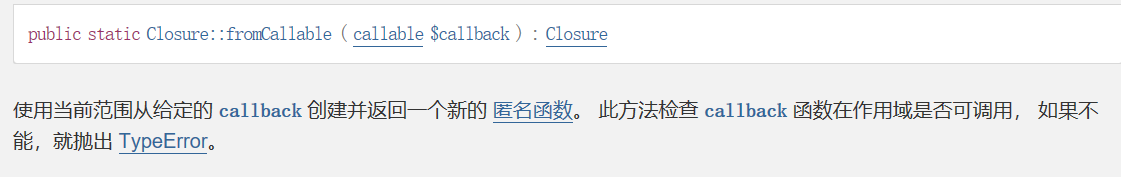
以上例程会输出：

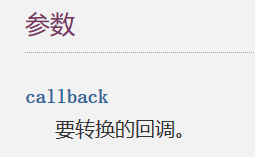
int(7)

int(8)

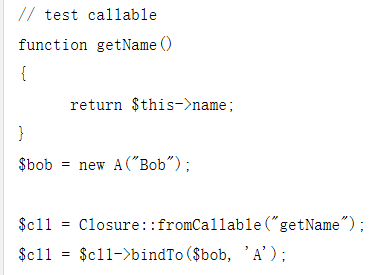
## Closure::fromCallable

Closure::fromCallable — 将 callable 转换为闭包





返回新创建的 [Closure](mk:@MSITStore:C:\\Users\\张卫航\\Desktop\\php_enhanced_zh.chm::/res/class.closure.html)， 或者如果 callback 在当前作用域无法调用， 则抛出 [TypeError](mk:@MSITStore:C:\\Users\\张卫航\\Desktop\\php_enhanced_zh.chm::/res/class.typeerror.html)。



# bindTo()跟bind()的差别

bindTo()跟bind()的差别只是用法不一样。

$bcl1 = Closure::bind($cl1, null, 'A');

$cl = function() { return $this->val; };

$cl = $cl->bindTo($ob2);