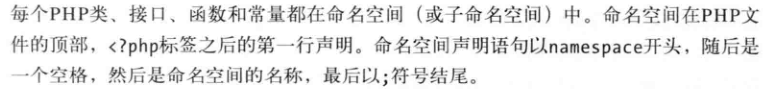
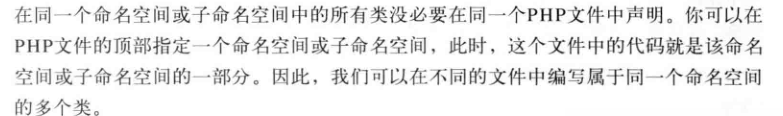
命名空间只是PHP语言的一种记号，PHP解释器会将其作为前缀加到类、接口函数和常量的名称前。

**声明命名空间**





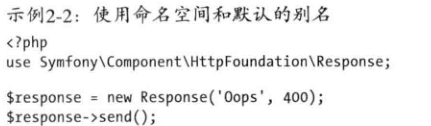
**导入类（use）**

没有使用use导入类时，我们可以这样使用其他类



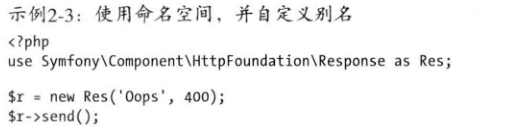
注意，此时要加上命名空间前面的\，而且每次实例化时都内写完整的命名空间

使用use 导入类



注意，使用use导入后只需要输入一次完全限定的类名，随后的实例化不用再输入完整的类名。

使用as指定类的别名，（不使用as默认别名就是类名）



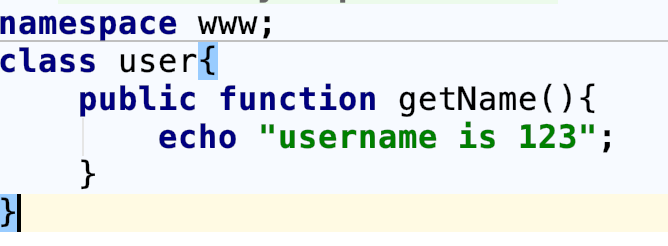
注意：应该在PHP文件顶部使用use关键字导入代码，而且要放到命名空间声明语句之后。Use关键字必须在全局作用域中，不能在类或者函数里。

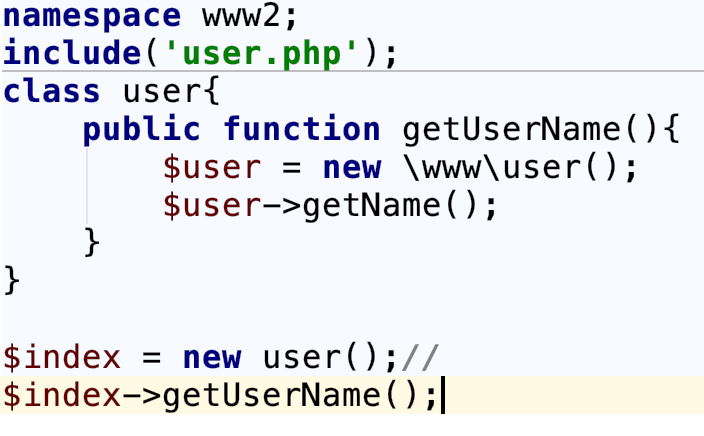
使用use关键字时无需在开头加上\符号，因为PHP假定导入的是完全限定的命名空间。

**命名空间和include()**

注意，命名空间只是为了降低不同文件重名的冲突，相当于类、常量、方法的前缀，**include()和require是引入文件，两种概念。**

例如下面的例子，相同类名，使用命名空间区分：





**全局命名空间**

有些引用的类没有命名空间，PHP会默认在当前命名空间中，此时会找不到引入的类，实例化时错误，例如PHP原生类Exception，需要在类的前面加上\前缀，这么做为了告诉PHP别在当前命名空间中查找，要在全局命名空间中查找，

<?php

namespace Test;

try {

　　throw new \Exception("抛出异常");//不加\会报错

} catch (\Exception $e) {

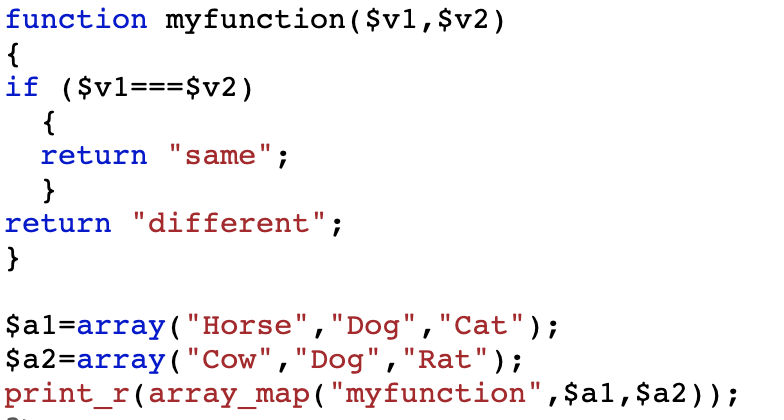
　　echo '捕获到异常'.$e->getMessage();

}

知识点：array\_map(‘myfunction’,array1,array2,array3...)

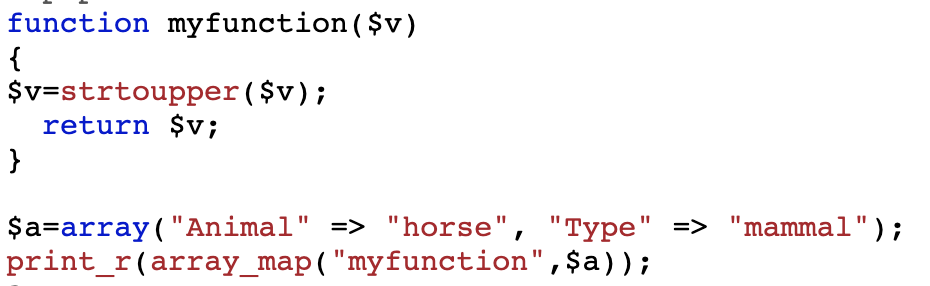
array\_map作用将数组里的每个值作为参数带入自定义函数，函数参数和数组个数对应，

例如1，比较两个数组的值



输出结果：Array ( [0] => different [1] => same [2] => different )

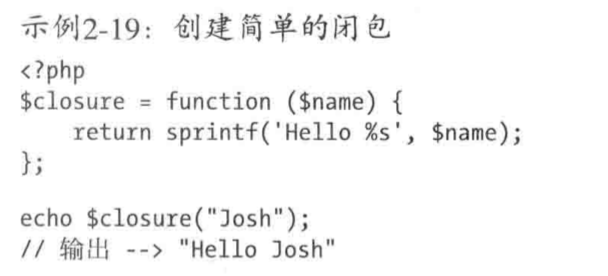
例如2:将数据值小写转大写



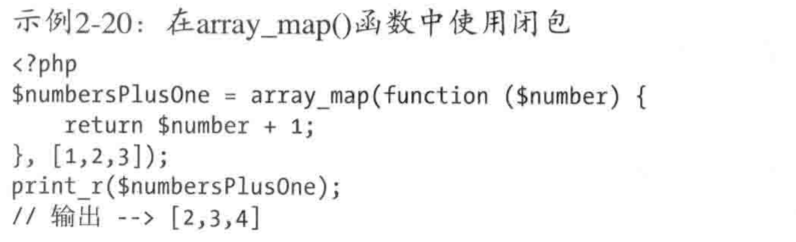
输出结果：Array ( [Animal] => HORSE [Type] => MAMMAL )

闭包

闭包和函数很像，使用一个匿名函数定义给一个变量：



通常把PHP闭包当作函数和方法的回调使用，**闭包可以作为参数传入其他PHP函数**，例如在



PHP-FIG

PHP-FIG的使命是实现框架的互操作性，通过接口、动加载机制和标准，让框架互相合作

59