<?php // 定义代码在 'MyProject' 命名空间中

namespace MyProject;

// ... 代码 ...

<?php namespace MyProject;

const CONNECT\_OK = 1;class Connection { /\* ... \*/ }function connect() { /\* ... \*/ }

namespace AnotherProject;

const CONNECT\_OK = 1;class Connection { /\* ... \*/ }function connect() { /\* ... \*/ }?>

<?php

namespace MyProject {

const CONNECT\_OK = 1;

class Connection { /\* ... \*/ }

function connect() { /\* ... \*/ }}

namespace AnotherProject {

const CONNECT\_OK = 1;

class Connection { /\* ... \*/ }

function connect() { /\* ... \*/ }}?>

namespace { // 全局代码

session\_start();

$a = MyProject\connect();

echo MyProject\Connection::start();}

<?php

namespace MyProject\Sub\Level; //声明分层次的单个命名空间

const CONNECT\_OK = 1;class Connection { /\* ... \*/ }function Connect() { /\* ... \*/ }

?>

## 使用命名空间：别名/导入

<?php

namespace foo;

use My\Full\Classname as Another;

// 下面的例子与 use My\Full\NSname as NSname 相同

use My\Full\NSname;

// 导入一个全局类

use \ArrayObject;

$obj = new namespace\Another; // 实例化 foo\Another 对象

$obj = new Another; // 实例化 My\Full\Classname　对象

NSname\subns\func(); // 调用函数 My\Full\NSname\subns\func

$a = new ArrayObject(array(1)); // 实例化 ArrayObject 对象// 如果不使用 "use \ArrayObject" ，则实例化一个 foo\ArrayObject 对象

?>

## 全局空间

如果没有定义任何命名空间，所有的类与函数的定义都是在全局空间