PHP 命名空间可以解决以下两类问题：

用户编写的代码与PHP内部的类/函数/常量或第三方类/函数/常量之间的名字冲突。

为很长的标识符名称(通常是为了缓解第一类问题而定义的)创建一个别名（或简短）的名称，提高源代码的可读性。

定义命名空间

默认情况下，所有常量、类和函数名都放在全局空间下，就和PHP支持命名空间之前一样。

命名空间通过关键字namespace 来声明。如果一个文件中包含命名空间，它必须在其它所有代码之前声明命名空间。语法格式如下；

<?php

// 定义代码在 'MyProject' 命名空间中

namespace MyProject;

// …代码…

可以在同一个文件中定义不同的命名空间

命名空间使用

file1.php

<?php

namespace Foo\Bar\subnamespace;

const FOO = 1;

function foo() {}

class foo

{

static function staticmethod() {}

}

file2.php

<?php

namespace Foo\Bar;

include 'file1.php';

const FOO = 2;

function foo() {}

class foo

{

static function staticmethod() {}

}

/\* 非限定名称 \*/

foo(); // 解析为函数 Foo\Bar\foo

foo::staticmethod(); // 解析为类 Foo\Bar\foo ，方法为 staticmethod

echo FOO; // 解析为常量 Foo\Bar\FOO

/\* 限定名称 \*/

subnamespace\foo(); // 解析为函数 Foo\Bar\subnamespace\foo

subnamespace\foo::staticmethod(); // 解析为类 Foo\Bar\subnamespace\foo,

// 以及类的方法 staticmethod

echo subnamespace\FOO; // 解析为常量 Foo\Bar\subnamespace\FOO

/\* 完全限定名称 \*/

\Foo\Bar\foo(); // 解析为函数 Foo\Bar\foo

\Foo\Bar\foo::staticmethod(); // 解析为类 Foo\Bar\foo, 以及类的方法 staticmethod

echo \Foo\Bar\FOO; // 解析为常量 Foo\Bar\FOO

命名空间使用 use

<?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(); // 实例化 ArrayObject 对象