

① 命名空间概述

在不同命名空间下的类、函数、常量等可以重名，同一命名空间下不允许给一个东西重定义。

要在全局空间以外定义命名空间，必须在.php文件<?php下一行首先用namespace声明(除declare语句可以在其之前外)

④ 定义命名空间

```
<?php
```

```
declare(encoding = 'UTF-8');
```

```
namespace ns_1;
```

代码 \rightarrow 归入 ns-1 内

```
?>
```

```
<?php
```

\rightarrow 默认自动加, 建议省略

```
namespace (\)Myproject\ns_2;
```

代码 \rightarrow 默认归入 ns-2 内

namespace ns-3;

代码; \rightarrow 默认归 ns-3 内

? >

可以分号后全行缩进入上一个 namespace 空间, 但多个空间建议

② namespace ns {
 代码 } ; 来区分

但不建议在一个文件中声明多个 namespace

③ 使用 命令空间.

不建议在一个文件中声明多个命令

空间, 但可以使用(导入)外部命名空间内容.

```
<?php
```

```
namespace foo;
```

```
use Full\Ns-1, Full\Ns-2\class;
```

代码;

use 只能是指 namespace 名或

classname, 常量、变量、函数

第2个在调用时加入 namespace

→ ~~~~~

来跟定

<?php

namespace foo;

use B\D, C\E as F

// 可能是ns,也可能是classname

F指代 C\E, 可能是ns和classname

// 以下为调用函数时解析情况

foo(); //调用 foo 中的 foo() 函数,

若无, 则调用全局函数

\foo(); 直接调用全局函数

my\foo(); 调用foo\my中的foo();

//不会尝试\my\foo(), 因为没有use my.

//以下为类创建

new B(); //创建foo\B类的对象

new D(); //创建B\D类对象

//因use: B\D中D此时被当作类名

new F(); //创建C\F类对象

1.1.1 使用

// 1. use C++ as 1. 中 1. 时 1. 作 1. 架 1. 公

new B(); 创建全局 B 类的对象

...

总之：

对于 \ 开头的即完全限定名调用，
编译时解，空间明确。

对于非限定调用，即只有一个

函数、类名若无任何 \ 的；解析
首先查找 use 中是否有符合的，
若无再看 namespace 中是否有，若无
则当作全局的（全局类或函数调用）

对非完全限定类和函数，都
在其前加上 namespace 再查找。

对函数：

$\text{foo}(); \rightarrow \begin{cases} 1. \text{ ns 查} \\ 2. \text{ 全局查} \end{cases}$

$\text{my}(\text{foo}); \rightarrow \text{ns}(\text{my}(\text{foo})) \text{ 查}$

$\backslash \text{my}(\text{foo}); \rightarrow \backslash \text{my}(\text{foo}) \text{ 查}$

对类:

$\text{new B}(); \begin{cases} 1. \text{ 导入规则查} \\ 2. \text{ ns 查} \end{cases}$

new my\Bc) ; \rightarrow ns\my\Bc) $\overline{\text{查}}$

new \Bc) ; \rightarrow \Bc) $\overline{\text{查}}$

{非全局限定符加上 namespace
{全局限定符直接查。
} \rightarrow 编译时完成

非限定符在运行时解析,

... ② ③

函数必须先 namespace 且, 一般
全局查 (因 use 不能指函数名,
所以不存在查导入规则)

类①首先导入规则查 ②再
namespace 查 (全局类必须用
\ 类名调用, 所以也不存在查
是否是全局的可能)