



NAMESPACE

命名空間 (namespace)

■ 目的：

- 給類別 (全域函數也可以) 一個更高層級的名稱，避免載入外部檔案時，因類別名稱一樣而產生錯誤

```
<?php // a.php
class Hello {
    function __toString() {
        return "Hello, AAA\n";
    }
}
```

```
<?php // b.php
class Hello {
    function __toString() {
        return "Hello, BBB\n";
    }
}
```

```
<?php // main.php
require 'a.php';
require 'b.php';
echo new Hello();
```

無法運作

加上命名空間

```
<?php // a.php
namespace Lib\A;
class Hello {
    function __toString() {
        return "Hello, AAA\n";
    }
}
```

```
<?php // b.php
namespace Lib\B;
class Hello {
    function __toString() {
        return "Hello, BBB\n";
    }
}
```

```
<?php // main.php
require 'a.php';
require 'b.php';
echo new Hello\B\Hello();
```

Use 語法

- as 為設定類別的別名，省略時，use 的最後一個項目為類別名稱

```
<?php // main.php  
require 'a.php';  
require 'b.php';
```

```
use Lib\A\Hello as AA;  
use Lib\B\Hello as BB;
```

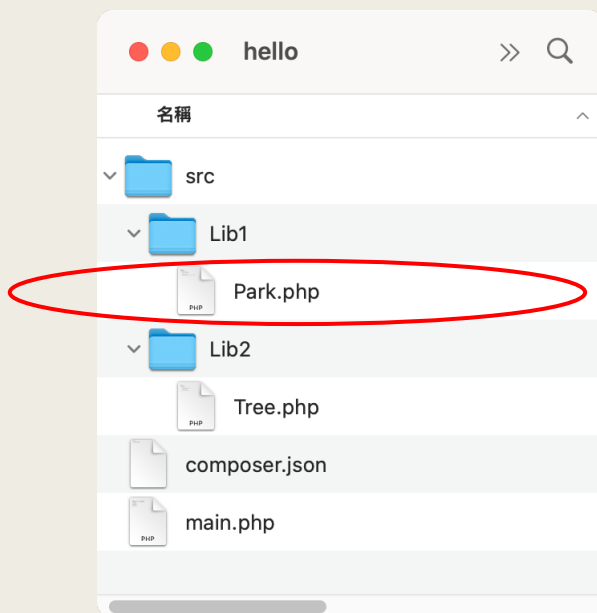
```
echo new AA();  
echo new BB();
```

COMPOSER 與 AUTOLOAD

psr-4 規範

■ 簡單來說

- 一個 class 一個檔案，檔名與 class 同，區分大小寫
- Namespace 對映到目錄名稱，例如



```
<?php
namespace Lib1;

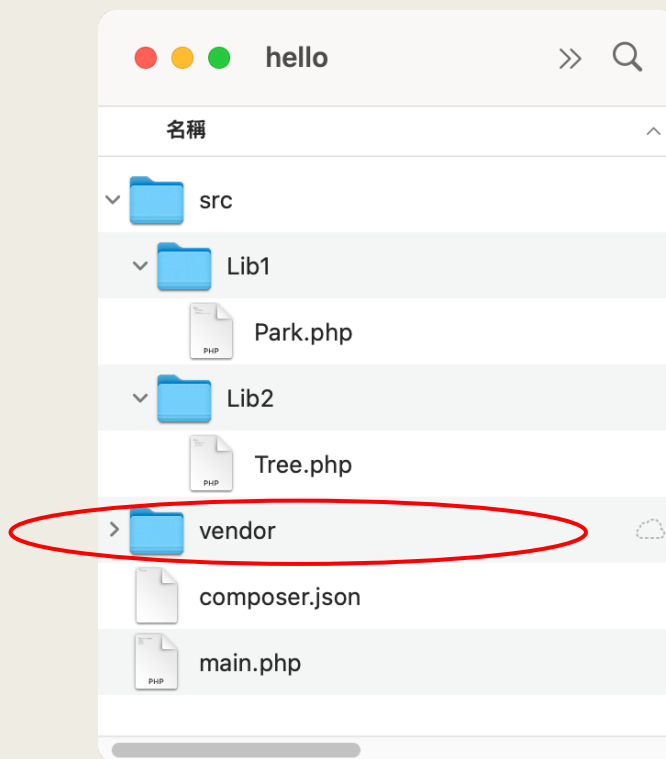
class Park {
    function __toString() {
        return "This is Park\n";
    }
}
```

composer.json 內容

```
{  
    "autoload": {  
        "psr-4": {  
            "Lib1\\": "src/Lib1/",  
            "Lib2\\": "src/Lib2/"  
        }  
    }  
}
```

產生 autoload.php 檔

- 將目前工作路徑切換到專案目錄，也就是 composer.json 所在的目錄中
- 執行指令
 - `composer dump-autoload`
- 完成後會產生 vendor 目錄
- autoload.php 在 vendor 中



main.php 的內容

```
<?php
require_once __DIR__ . '/vendor/autoload.php';

use Lib1\Park;
use Lib2\Tree;

echo new Park();
echo new Tree();
```

修改 class Tree

- 增加建構子

```
<?php
namespace Lib2;

class Tree {
    function __construct(int $count=0) {
        $this->count = $count;
    }

    function __toString() {
        return "This is Tree\n";
    }
}
```

在 Park 中引用 Tree

- 完全不需要使用 require 或 include 了

```
<?php
namespace Park;
use Lib2\Tree;

class Park {
    function __toString() {
        return "This is Park\n";
    }

    function info() {
        $tree = new Tree(10);
        echo "{$tree->count} trees in the park\n";
    }
}
```

修改 main.php

- 改一下 main.php 後執行看看

```
<?php
require_once __DIR__ . '/vendor/autoload.php';

use Lib1\Park;
use Lib2\Tree;

echo new Park();
echo new Tree();

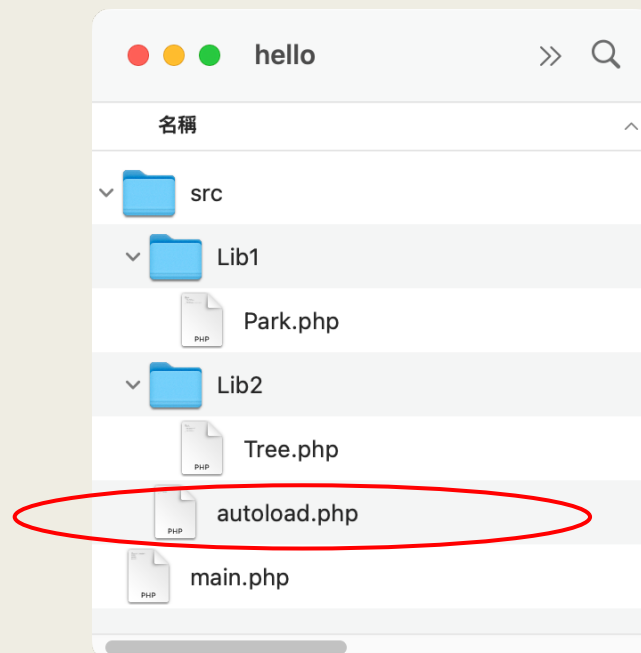
$park = new Park();
$park->info();
```

```
This is Park
This is Tree
10 trees in the park
```

spl_autoload_register()

autoload.php 內容

```
<?php
spl_autoload_register(function($className) {
    $file = dirname(__DIR__) . '\\src\\' . $className . '.php';
    $file = str_replace('\\', DIRECTORY_SEPARATOR, $file);
    // echo $file . "<br>\n";
    if (file_exists($file)) {
        include $file;
    }
});
```



main.php

```
<?php  
require_once __DIR__ . '/src/autoload.php';  
  
use Lib1\Park;  
use Lib2\Tree;  
  
echo new Park();  
echo new Tree();
```