Trong một dự án lớn có bao giờ bạn nghĩ đến trường hợp là sẽ xảy ra các class có trùng tên nhau không? Chính vì điều đó mà kể tử phiên bản **PHP 5.3** trở đi thì PHP có hỗ trợ cho chúng ta chức năng mới là namespace ...

**1, Namespace là  gì?**

- Namespace giúp tạo ra một không gian tên cho hàm và lớp trong lập trình nói chung và trong PHP nói riêng.

-Vì ở đây là series hướng đối tượng nên mình sẽ chỉ trình bày về namespace trong hướng đối tượng, trong hướng đối tượng thì namespace có tác dụng tạo ra định danh cho lớp một cách cụ thể hóa hơn.

**VD:**như chúng ta có 2 file mỗi file đều chứa một class và 2 class này lại có một điểm chung là trùng tên. Giờ đây khi bạn nhúng cả 2 file này vào và gọi class thì ngay lập tức chương trình sẽ báo lỗi. Để khắc phục điều đó thì chúng ta cần khai báo **namespace** cho hai class đó (đọc tiếp để xem cách khai báo).

**2, Khai báo namespace.**

-Để khai báo **namespace** trong PHP chú ta sử dụng cú pháp sau:

namespace Name;

 Trong đó: **Name** là tên của namespace.

**Chú ý:**khi khai báo **namespace** thì chúng ta phải đặt nó ở **phía trên cùng** của file. Hơn nữa nếu như bạn muốn đặt tên **namespace** đúng chuẩn thì hãy tham khảo series PHP standards.

**VD:**

class Package

{

public function sayHello()

{

echo 'Hello World!';

}

}

-Và bạn cũng có thể hoàn toàn đặt tên namespace theo các cấp được.

**VD**: Tạo một class HomeController trong thư mục app\controllers và đặt tên namespace như sau:

namespace App\Controllers;

class HomeController

{

//code

}

**3, Gọi namespace.**

- Khi mà một class đã được một namespace định danh thì bạn sẽ không thể gọi theo cách thông thường được nữa mà phải gọi với cú pháp:

new tenNamespace\tenClass();

**VD**:

-Tạo một file **ConNguoi.php** có nội dung như sau:

<?php

namespace ConNguoi;

class ConNguoi

{

private $name = 'Con Người';

public function getName()

{

return $this->name;

}

}

-Tiếp đó tạo một file **index.php** cùng cấp với file **ConNguoi.php** và lúc này chúng ta sẽ phải gọi class ConNguoi như sau:

<?php

include 'ConNguoi.php';

$connguoi = new ConNguoi\ConNguoi();

echo $connguoi->getName();

//kết quả: Con Người

**Nạp namespace bằng use**

-Nếu như bạn không muốn gọi namespace theo cách trên thì bạn sử dụng từ khóa use để nạp theo cú pháp sau:

use tenNamespace\tenClass;

**Chú ý**: Khai báo nạp namespace cũng phải được đặt trên đầu của một file, nếu như trong một file có sử dụng namespace và use thì **use** sẽ được đặt dưới namespace.

**VD**: Mình sẽ sử dụng use để nạp namespace của class ConNguoi, khi đó file index.php sẽ được viết như sau:

<?php

include 'ConNguoi.php';

use ConNguoi\ConNguoi;

$connguoi = new ConNguoi();

echo $connguoi->getName();

//kết quả: Con Người

**VD**: Một file mà có khai báo namespace và use thì sẽ viết như sau:

<?php

namespace nguoilon;

include 'ConNguoi.php';

use ConNguoi\ConNguoi;

class NguoiLon extends ConNguoi

{

//code

}

**4, Định danh cho namespace.**

-Giả sử **namespace** của chúng ta rất dài hoặc vì một lý do nào đó mà bạn không muốn gọi namespace như thế, thì PHP cũng có cung cấp cho chúng ta một phương pháp đó là **tạo định danh cho namespace**.

-Để tạo định danh mới cho namespace trong PHP chúng ta sử dụng từ khóa as với cú pháp như sau:

use tenNamespace as tenMoi;

**VD**: Tạo định danh cho namespace ConNguoi ở trên.

<?php

include 'ConNguoi.php';

use ConNguoi as People;

$connguoi = new People\ConNguoi();

echo $connguoi->getName();

//kết quả: Con Người

**5, Nhiều namespace trong một file.**

-Như ở trên mình có nói là khi khai báo namespace thì chúng ta phải đặt ở đầu file, nhưng nếu như ở trong một file có nhiều namespace thì chúng ta sẽ khai báo như sau:

namespace nameone;

class ClassOne

{

//

}

namespace nametwo;

class ClassTwo

{

//

}

**VD**:

-Tạo một file Class.php có nội dung như sau:

<?php

namespace Class\ConNguoi;

class ConNguoi

{

private $name = 'Con Người';

public function getName()

{

return $this->name;

}

}

namespace Class\NguoiLon;

class NguoiLon

{

private $name = 'Nguoi Lon';

public function getName()

{

return $this->name;

}

}

-Và chúng ta sẽ tạo một file index.php để gọi 2 class đó như sau:.

<?php

include 'ConNguoi.php';

$connguoi = new Class\ConNguoi\ConNguoi();

echo $connguoi->getName();

//kết quả: Con Người

$nguoilon = new Class\NguoiLon\NguoiLon();

echo $nguoilon->getName();

//kết quả: Nguoi Lon