Như đã được học từ trước thì các bạn cũng đã biết nếu muốn [khởi tạo một class](http://toidicode.com/lop-thuoc-tinh-phuong-thuc-trong-php-98.html) được khai báo trong một file khác thì chúng phải nhúng file đó vào trong file muốn khởi tạo thì mới có thể sử dụng được, nhưng nếu như chúng ta khởi tạo nhiều class trong một files mà các class đó được khai báo từ các file khác nhau thì sẽ phải include rất nhiều file. Điều đó sẽ khiến cho code của chúng ta trở lên rất rối và để khắc phục được điều đó thì chúng sẽ sử dụng hàm \_\_autoload() trong PHP.

**1, \_\_autoload() là gì?**

- \_\_autoload() là một hàm đặc biệt được PHP giới thiệu từ phiên bản 5.0, hàm \_\_autoload() này sẽ được gọi khi chúng ta **khởi tạo một đối tượng không xác định được**. Vì \_\_autoload() là một hàm nên chúng ta sẽ khai báo nó như một hàm bình thường.

**Cú Pháp**:

function \_\_autoload($clasName)

{

//code

}

**Trong đó**: $className là biến mà các bạn truyền vào để lấy ra tên class(tên biến là gì cũng được nhưng bắt buộc phải khai báo).

-Và để chứng minh cho câu nói: '*nó được gọi khi chúng ta khởi tạo một đối tượng không xác định được*' thì các bạn xem ví dụ sau:

+Đầu tiên chúng ta khai báo hàm\_\_autoload và classConNguoi trên cùng một file và đồng thời khởi tạo luôn class ConNguoi.

<?php

//Khai báo hàm \_\_autoload

function \_\_autoload($className)

{

echo 'Bạn vừa khởi tạo class: ' . $className;

}

//Khai báo class ConNguoi

class ConNguoi

{

//Khai báo hàm khởi tạo

public function \_\_construct()

{

echo 'Class ConNguoi';

}

}

//Khởi tạo class ConNguoi

$connguoi = new ConNguoi();

//Kết Quả: Class ConNguoi

+Như các bạn đã thấy thì hàm \_\_autoload() **không được gọi khi chúng ta khởi tạo class được xác định**. Giờ mình thử xóa bỏ đoạn khai báo class ConNguoi nhưng vẫn khởi tạo nó xem sao.

<?php

//Khai báo hàm \_\_autoload

function \_\_autoload($className)

{

echo 'Bạn vừa khởi tạo class: ' . $className;

}

//Khởi tạo class ConNguoi

$connguoi = new ConNguoi();

//Kết Quả:

//Bạn vừa khởi tạo class: ConNguoi

// Fatal error: Class 'ConNguoi' not found

-Oh, Vậy là hàm \_\_autoload() đã được gọi. Như vậy câu nói trên là đúng nhé!

**2, Các Ví Dụ.**

-Vì hàm \_\_autoload() sẽ tự động được gọi khi chúng ta khởi tạo một đối tượng không xác định được, nên nó được ứng dụng vào việc tạo ra các ứng dụng **autoloading classes**.

**Autoload class thông thường.**

**VD1**: Chúng ta có 3 file NguoiLon.php, TreCon.php và index.php ở cùng cấp thư mục.

+File NguoiLon.php, chứa class NguoiLon có nội dung như sau:

<?php

class NguoiLon

{

public function \_\_construct()

{

echo 'Class NguoiLon';

}

}

+File TreCon.php, chứa class TreCon có nội dung như sau:

<?php

class TreCon

{

public function \_\_construct()

{

echo 'Class TreCon';

}

}

+Thông thường, để khởi tạo 2 class trên trong file index.php thì chúng ta phải include từng file class như sau:

<?php

//Nhúng file ConNguoi

include\_once 'NguoiLon.php';

//Nhúng file ConNguoi

include\_once 'TreCon.php';

//Khởi tạo 2 class

$nguoilon = new NguoiLon();

//Kết Quả: Class NguoiLon

$trecon = new TreCon();

//Kết Quả: Class TreCon

+Nhưng nếu chúng ta sử dụng  hàm \_\_autoload() thì sẽ chỉ phải khai báo ngắn gọn như sau:

<?php

//khai báo hàm \_\_autoload

function \_\_autoload($className)

{

//kiểm tra xem file tồn tại không

if(file\_exists($className . '.php')){

//Nếu tồn tại thì nhúng file vào.

include\_once $className . '.php';

}

}

//Khởi tạo 2 class

$nguoilon = new NguoiLon();

//Kết Quả: Class NguoiLon

$trecon = new TreCon();

//Kết Quả: Class TreCon

**Autoload khi gọi phương thức tĩnh.**

-Và hàm \_\_autoload() này cũng được gọi khi chúng ta gọi phương thức tĩnh không xác định.

**VD**: Chúng ta có 2 file ConNguoi.php và một file index.php ở cùng thư mục.

+File ConNguoi.php, chứa class ConNguoi có nội dung như sau:

<?php

class ConNguoi

{

private static $name = 'ConNguoi';

public function \_\_construct()

{

}

public static function getName()

{

echo static::$name;

}

}

+File index.php, có nội dung như sau:

<?php

//khai báo hàm \_\_autoload

function \_\_autoload($className)

{

//kiểm tra xem file tồn tại không

if(file\_exists($className . '.php')){

//Nếu tồn tại thì nhúng file vào.

include\_once $className . '.php';

}

}

ConNguoi::getName();

//Kết Quả: ConNguoi

-Sau khi chạy file **index.php**thì như các bạn cũng đã thấy là nó vẫn chạy bình thường.