**1, Tổng quan.**

-Dựa vào bài [thành phần tĩnh trong PHP](http://toidicode.com/static-va-final-trong-php-huong-doi-tuong-105.html) thì chúng ta cũng đã biết về self rồi, ở đây chúng ta tạm hiểu self dùng để gọi các thành phần tĩnh trong class.

-Còn $this thì từ đầu series đến giờ chúng ta cũng dùng rất là nhiều, có thể hiểm nôm na là $this dùng để gọi các thành phần thường( không phải là tĩnh) trong class.

-Nhưng ở bài này chúng ta sẽ cùng nhau đi tìm hiểu về sự khác nhau sâu xa giữa 2 từ khóa này như: $this có gọi được phương thức,thuộc tính tĩnh không, self có gọi được thuộc tính, phương thức thường không,...

-Và để cho dễ hiểu hơn thì mình sẽ tạo ra 1 ví dụ mà sẽ áp dụng nó cho cả bài này.

**VD**: Mình có 1 class ConNguoi có 2 phương thức getClass(),echoClass() và class NguoiLon kế thừa class ConNguoi và đồng thời override lại phương thức getclass() của class ConNguoi.

<?php

// Khai báo class ConNguoi

class ConNguoi

{

public function getClass()

{

return 'Đây là class ConNguoi';

}

public function echoClass()

{

echo $this->getClass();

}

}

//Khai báo class NguoiLon

class NguoiLon extends ConNguoi

{

public function getClass()

{

return 'Đây là class NguoiLon';

}

}

**2, Sự khác nhau.**

-Sau khi đã tạo ra được các tài nguyên để tìm hiểu thì giờ chúng ta cùng bắt tay vào phân tích nào.

**$this và self trong đối tượng**

-Đầu tiên chúng ta sẽ khởi tạo class **NguoiLon** và gọi tới phương thức echoClass() như từ trước tới giờ chúng ta vẫn gọi nhé.

<?php

// Khai báo class ConNguoi

class ConNguoi

{

public function getClass()

{

return 'Đây là class ConNguoi';

}

public function echoClass()

{

echo $this->getClass();

}

}

//Khai báo class NguoiLon

class NguoiLon extends ConNguoi

{

public function getClass()

{

return 'Đây là class NguoiLon';

}

}

//Khởi tạo class

$nguoilon = new NguoiLon();

//echoClass

$nguoilon->echoClass();

//Kết Quả : Đây là class NguoiLon

- Kết quả đúng như những gì chúng ta đã được học luôn nhé. Giờ chúng ta thử thay đổi $this trong phương thức echoClass() thành self và xem kết quả thế nào.

<?php

// Khai báo class ConNguoi

class ConNguoi

{

public function getClass()

{

return 'Đây là class ConNguoi';

}

public function echoClass()

{

//Thay đổi bằng self

echo self::getClass();

}

}

//Khai báo class NguoiLon

class NguoiLon extends ConNguoi

{

public function getClass()

{

return 'Đây là class NguoiLon';

}

}

//Khởi tạo class

$nguoilon = new NguoiLon();

//echoClass

$nguoilon->echoClass();

//Kết Quả : Đây là class ConNguoi

- Như các bạn đã thấy thì kết quả đã thay đổi hoàn toàn. Từ đó chúng ta sẽ đưa ra các kết luận tại thời điểm hiện tại như sau:

+ **Kết luận đầu tiên**: Chúng ta có thể thấy là self hoàn toàn có thể gọi được phương thức bình thường.

+ **Kết luận thứ hai**: Dựa vào kết quả của 2 ví dụ thì chúng ta kết luận là self sẽ **gọi đến đối tượng khai báo nó**, còn $this sẽ **gọi đến đối tượng hiện tại**.

**$this và self trong thành phần tĩnh**

-Về phần self trong thành phần tĩnh thì mình đã trình bày quá nhiều trong bài [thành phần tĩnh trong PHP rồi](http://toidicode.com/static-va-final-trong-php-huong-doi-tuong-105.html), nên ở đây mình không nói đến nữa mà chúng ta sẽ chỉ ví dụ trên **$this** thôi.

**$this trong phương thức tĩnh**

**VD**: Chúng ta sẽ sử dụng $this trong phương thức tĩnh để gọi các thành phần thường.

<?php

// Khai báo class ConNguoi

class ConNguoi

{

private $name = 'ConNguoi';

public static function setName($name)

{

$this->name = $name;

}

public function getName()

{

return $this->name;

}

public static function echoName()

{

echo $this->getName();

}

}

//Khởi tạo class

$nguoilon = new ConNguoi();

//setName

$nguoilon->setName('Vũ Thanh Tài');

//Kết Quả : Using $this when not in object context

// echoName

$nguoilon->echoName();

//Kết Quả : Using $this when not in object context

-Như các bạn đã thấy, khi mà đúng ở trong phương thức tĩnh thì chúng ta không thể gọi được các thành phần thường, hay nói một cách khác là **$this không sử dụng được trong phương thức tĩnh**.

**$this gọi thuộc tính tĩnh**

**VD**: Chúng ta sẽ tạo thêm thuộc tính tĩnh name cho class ConNguoi và gọi nó trong phương thức getClass() như sau:

<?php

// Khai báo class ConNguoi

class ConNguoi

{

public static $name = 'ConNguoi';

public function setName($name)

{

$this->name = $name;

}

public function echoName()

{

echo $this->name;

}

}

//Khởi tạo class

$nguoilon = new ConNguoi();

//echoName

$nguoilon->echoName();

//Kết Quả : Undefined property: NguoiLon::$name

echo $nguoilon->name;

//Kết Quả: Undefined property: NguoiLon::$name

-Như vậy các bạn đã thấy chương trình đã báo lỗi. **Nhưng khoan đã** rõ ràng là trong class của chúng ta có khai báo thuộc tính **name** mà, vậy lý do sao nhỉ?. Thực ra **name đó không phải là name** mà các bạn khai báo là **thuộc tính tĩnh** trong class đâu mà **bản chất thì chúng ta không hề gọi được thuộc tính tĩnh bằng $this**.

'*Đệch! các ông không tin tôi đúng không?, tôi sẽ chứng minh cho các ông xem.*'

<?php

// Khai báo class ConNguoi

class ConNguoi

{

public static $name = 'ConNguoi';

public function setName($name)

{

$this->name = $name;

}

public function echoName()

{

echo $this->name;

}

}

//Khởi tạo class

$nguoilon = new ConNguoi();

//setName

$nguoilon->setName('Vũ Thanh Tài');

//echoName

$nguoilon->echoName();

//Kết Quả : Vũ Thanh Tài

echo $nguoilon->name;

//Kết Quả: Vũ Thanh Tài

// Khởi tạo tiếp class Tre con

$trecon = new ConNguoi();

//gọi phương thức echoName

$trecon->echoName();

//Kết Quả: Undefined property: ConNguoi::$name

//gọi thuộc tính name

$trecon->name;

//Kết Quả: Undefined property: ConNguoi::$name

-Như các bạn thấy đó, nếu thuộc tính **name** kia là thuộc tính tĩnh thì nó phải được lưu lại giá trị cuối cùng chứ! (ai chưa biết về static thì xem [link](http://toidicode.com/static-va-final-trong-php-huong-doi-tuong-105.html)). Tin tôi chưa?

**$this gọi phương thức tĩnh.**

**VD:**Tạo một class ConNguoi có một phương thức tĩnh getStatic() và một phương thức thường echoStatic() như sau:

<?php

// Khai báo class ConNguoi

class ConNguoi

{

public static function getStatic()

{

return 'Đây là phương thức getStatic';

}

public function echoStatic()

{

echo $this->getStatic();

}

}

$connguoi = new ConNguoi();

$connguoi->echoStatic();

//Kết quả: Đây là phương thức getStatic

-Như các bạn thấy chương trình chạy tốt không có lỗi gì. Chúng ta hoàn toàn có thể dựa vào đó mà kết luận được như sau: **$this có thể gọi được phương thức tĩnh.**