**1, Lambda.**

**Lambda là gì?**

- **Lambda** là một anonymous function (**hàm ẩn danh**) nó có thể **khai báo,định nghĩa ở bất kỳ đâu** và **không có khả năng tái sử dụng**.

- **Lambda** chỉ tồn tại trong phạm vi của biến mà nó được định nghĩa, vì vậy nếu như biến đó vượt ra ngoài phạm vi thì hàm này cũng không còn tác dụng nữa.

- **Lambda** thường được dùng để gán vào biến, hay được gán vào hàm,class như một tham số.

**Khai báo lambda**

-Để khai báo lambda trong PHP chúng ta sử dụng cú pháp:

function (argument)

{

//code

}

Hoặc có thể sử dụng hàm create\_function() trong PHP.

create\_function('', argument);

**Trong đó**: Trong đó argument là các tham số bạn muốn truyền vào tham số ẩn danh.

**VD1**: Khai báo hàm ẩn có tác dụng in ra chuỗi '*Học lập trình online toidicode.com*'.

function ()

{

return 'Học lập trình online toidicode.com';

}

**Sử dụng lambda**

-Như ở trên mình có nói là lambda thường được gán vào biến, hoặc gán vào hàm, class như một tham số. Giờ mình sẽ làm các ví dụ chứng minh cho điều đó.

**VD2**: Sử dụng hàm ẩn danh vào biến.

<?php

//gán biến bằng hàm ẩn danh.

$lambda = function () {

return 'Học lập trình online toidicode.com';

};

//gọi biến

echo $lambda();

**Chú ý**: Nếu gán hàm ẩn danh bạn phải chú ý dấu ; phía sau định nghĩa hàm ẩn danh.

**VD3**: Sử dụng hàm ẩn danh như một tham số trong hàm.

<?php

function getRole($role)

{

return $role();

}

echo getRole(function () {

return 'Học lập trình online toidicode.com';

});

**VD4**: Sử dụng hàm ẩn danh như một tham số trong class, phương thức.

<?php

class Role

{

public $role;

public function \_\_construct($role)

{

$this->role = $role;

}

function getRole($role)

{

return $role();

}

}

//khởi tạo class Role

$role = new Role(function () {

return 'Học lập trình PHP Toidicode.com';

});

//gán data bằng thuộc tính role

$data = $role->role;

//hiển thị role

echo $data();

//Kết quả: Học lập trình PHP Toidicode.com

//gọi getRole

echo $role->getRole(function () {

return 'Học lập trình Nodejs Toidicode.com';

});

//Kêt quả: Học lập trình Nodejs Toidicode.com

**Truyền tham số trong lambda**

-Đương nhiên, vì lambda cũng là hàm nên phải cho truyền tham số vô rồi.

**VD5**: Truyền tham số vào hàm ẩn danh.

<?php

$content = function ($message) {

return 'Content: ' . $message;

};

echo $content('Học Lập trình');

**2, Closure.**

**Closure là gì?**

-Thực ra thì một **closure cũng là một lambda**, nhưng closure có thêm chức năng là có thể **sử dụng các biến bên ngoài phạm vi mà nó được tạo ra**.

**Khai báo closure**

-Để khai báo closure trong PHP chúng ta sẽ sự dụng cú pháp:

function (argument) use (scope) {

//code

}

**Trong đó**:

* **argument** là các tham số các bạn muốn truyền vào trong closure.
* **scope** là danh sách các biến phía ngoài closure mà các bạn muốn sử dụng trong closure.

**VD6**:

<?php

$scope = 'Vũ Thanh Tài';

$name = function () use ($scope) {

return $scope;

};

echo $name();

//Kết quả: Vũ Thanh Tài

**Sử dụng closure**

- Vì closure khá giống với lambda nên mình sẽ không ví dụ các phần giống, mà sẽ chỉ trình bày các VD khác liên quan.

**VD7**: Tham chiếu trong Closure.

<?php

$number = 1;

$getNextNumber = function () use (&$number) {

return ++$number;

};

$getPreNumber = function () use (&$number) {

return --$number;

};

echo $getNextNumber();

//kết quả: 2

echo $number;

// Kết quả: 2

echo $getPreNumber();

//Kết quả: 1

**VD8**: Closure trong Class.

<?php

class Cart

{

const PRICE\_BUTTER = 1.00;

const PRICE\_MILK = 3.00;

const PRICE\_EGGS = 6.95;

protected $products = array();

public function add($product, $quantity)

{

$this->products[$product] = $quantity;

}

public function getQuantity($product)

{

return isset($this->products[$product]) ? $this->products[$product] :

FALSE;

}

public function getTotal($tax)

{

$total = 0.00;

$callback =

function ($quantity, $product) use ($tax, &$total) {

$pricePerItem = constant(\_\_CLASS\_\_ . "::PRICE\_" .

strtoupper($product));

$total += ($pricePerItem \* $quantity) \* ($tax + 1.0);

};

array\_walk($this->products, $callback);

return round($total, 2);

}

}

$my\_cart = new Cart;

// Thêm sản phẩm

$my\_cart->add('butter', 1);

$my\_cart->add('milk', 3);

$my\_cart->add('eggs', 6);

// In ra tổng số tiền khi đã sale 5%.

print $my\_cart->getTotal(0.05) . "\n";

// Kết Quả: 54.29