

# LẬP TRÌNH WEB 1

## | Web Development 1 |

*Nguyễn Huy Hoàng – Bùi Thị Phương Thảo – Phan Thanh Nhuan*

*[02 . 2018]*

*Dùng kèm Giáo trình Lập trình web 1, Khoa Công nghệ thông tin, Cao đẳng Công nghệ Thủ Đức*



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY



Bài 2.

# LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG PHP



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY





## 2.1 CLASS



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY



# Khai báo class

Để khai báo một class, ta dùng từ khóa **class**.

```
class ClassName
{
    // Các properties
    public $var = 'a default value';

    // Các methods
    public function displayVar() {
        echo $this->var;
    }
}
```

# Tên class

- Phải được bắt đầu bằng ký tự hoặc dấu \_
- Không được bắt đầu bằng số
- Có thể chứa ký tự A-z, ký số 0-9 hoặc dấu \_
- Được Viết Hoa các ký tự đầu tiên của mỗi từ



## \$this->

- \$this là một biến tham chiếu đến chính đối tượng được gọi.
- Để gọi một property hay method của class, ta dùng dấu -> trước tên property hay method cần gọi.



## 2.2 **NEW**



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY



# Tạo instance

- Để tạo ra một thể hiện (instance) của một class, ta dùng từ khóa **new**.
- Nếu một chuỗi chứa tên lớp được sử dụng với **new**, một instance của class cũng được tạo ra.

```
$instance = new SimpleClass();
```

```
// Tạo instance với một biến:
```

```
$className = 'SimpleClass';
```

```
$instance = new $className(); // tương đương  
new SimpleClass()
```





2.3

# MAGIC METHODS



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY



# Magic methods

- Magic methods (hoặc special methods) là các hàm được cung cấp sẵn.
- Thực hiện các công việc thông thường hay được gọi bên trong các đối tượng.
- `__construct()`, `__destruct()`, `__call()`, `__callStatic()`, `__get()`, `__set()`, `__isset()`, `__unset()`, `__sleep()`, `__wakeup()`, `__toString()`, `__invoke()`, `__set_state()`, `__clone()`, `__debugInfo()`

Xem thêm tại: <http://php.net/manual/en/language.oop5.magic.php>



# Constructors

- Hàm thực hiện các công việc khi một instance được tạo ra.
- Magic method của constructors là **\_\_construct()**.

`__CLASS__` trả về tên của class được gọi. Hằng này là một magic constant được PHP cung cấp sẵn.

```
class MyClass
{
    public $prop1 = "I'm a class property!";
    public function __construct() {
        echo 'The class ', __CLASS__, ' was initiated!<br />';
    }
    public function setProperty($newval) {
        $this->prop1 = $newval;
    }
    public function getProperty() {
        return $this->prop1 . "<br />";
    }
}
```

# Destructors

- Hàm thực hiện các công việc khi một instance bị hủy.
- Magic method của destructors là **`__destruct()`**.

```
class MyClass
{
    // ... tiếp theo ví dụ trên.
    public function __destruct() {
        echo 'The class "', __CLASS__, '" was destroyed.<br />';
    }
}
```

# ▮ toString()

Để tránh xảy ra lỗi khi có một dòng code nào đó xuất một class ra dưới dạng chuỗi, ta dùng magic method **\_\_toString()**.

```
class MyClass
{
    // ... tiếp theo ví dụ trên.
    public function __toString() {
        echo "Using the toString method: ";
        return $this->getProperty();
    }
}
```

## 2.4 KẾ THỪA CLASS TRONG PHP



# Kế thừa class

- Các class có thể kế thừa properties và methods của một lớp khác bằng cách sử dụng từ khóa **extends**.

```
class MyOtherClass extends MyClass
{
    public function newMethod() {
        echo "From a new method in " . __CLASS__ . "<br />";
    }
}
```

# Overwrite

- Để thay đổi một property hay method kế thừa từ class cha, ta chỉ cần đơn giản ghi đè bằng cách định nghĩa lại property hay method đó trong class con.

```
class MyOtherClass extends MyClass
{
    public function __construct() {
        echo "A new constructor in " . __CLASS__ . "<br />";
    }

    public function newMethod() {
        echo "From a new method in " . __CLASS__ . "<br />";
    }
}
```



# Overwrite

- Khi có nhu cầu giữ lại hàm gốc từ class cha, đồng thời thêm vào đó một vài câu lệnh khác, ta dùng **parent::** trước tên hàm cha cần giữ lại.

```
class MyOtherClass extends MyClass
{
    public function __construct() {
        parent::__construct(); // Gọi __construct() của lớp cha
        echo "A new constructor in " . __CLASS__ . "<br />";
    }

    public function newMethod() {
        echo "From a new method in " . __CLASS__ . "<br />";
    }
}
```



## 2.5 **VISIBILITY**



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY



# Visibility

- Properties và methods trong PHP phải được gán visibility: **public**, **protected** và **private**.
- Property hay method còn có thể được xác định là **static** – cho phép chúng có thể được gọi mà không cần khởi tạo instance.

# Public / Private / Protected

- **Public** properties hay methods có thể được truy cập ở bất kỳ nơi đâu, bao gồm cả bên trong và bên ngoài class.
- **Private** properties hay methods chỉ có thể được truy cập bên trong nội bộ class.
- **Protected** properties hay methods chỉ có thể được truy cập bên trong class đó hoặc trong các class con kế thừa từ class đó.

# Static

- Khi properties hay methods được khai báo kèm từ khóa **static**, chúng có thể được truy cập mà không cần khởi tạo instance cho class đó.
- Các gọi static properties hay methods dùng dấu `::` (scope resolution operator):

TênClass::tênProperty

TênClass::tênMethod

# Static

- Một trong những ưu điểm chính của static properties là dữ liệu của chúng được lưu giữ trong suốt quá trình hoạt động.

```
class MyClass
{
    //... các properties cũ giống như các ví dụ trên
    public static $count = 0;

    //... các methods cũ giống như các ví dụ trên
    public static function plusOne() {
        return "The count is " . ++self::$count . "<br />";
    }
}

//Gọi hàm plusOne mà không cần khởi tạo instance cho
MyClass
echo MyClass::plusOne();
```



2.6

# AUTOLOADING



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY



# Autoloading

- Khi lập trình hướng đối tượng với PHP, ta thường tạo mỗi class là một file .php → phải liên kết tất cả các file này lại với nhau.
- Hàm `spl_autoload_register()` hỗ trợ load các file class.php một cách tự động, miễn là chúng ta đặt tên file thống nhất.
- Hiện nay có các tiêu chuẩn đặt tên file và cấu trúc thư mục chứa class, ví dụ như PSR-4 (<https://www.php-fig.org/psr/psr-4/>)



# Autoloading

Ví dụ chúng ta có hai class là MyClass1 và MyClass2 được định nghĩa lần lượt trong hai file MyClass1.php and MyClass2.php

```
spl_autoload_register(function ($class_name) {  
    require $class_name . '.php';  
});  
$obj = new MyClass1();  
$obj2 = new MyClass2();
```



2.7

# DOCBLOCKS



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY



# DocBlock

- DocBlock là một chuẩn ghi chú được sử dụng rộng rãi dùng để ghi chú lại các thông tin cho class.
- Phần thông tin ở giữa sẽ chứa tên class, mô tả về class, và các tag được định nghĩa sẵn của DocBlock.

```
/**  
 * This is a very basic DocBlock  
 */
```

# DocBlock tags

- @author: Tên tác giả của class. Nếu có nhiều tác giả, có thể sử dụng nhiều @author. Định dạng để ghi @author là:

*@author Tên Đây Đủ <email>*

- @copyright: Thông tin bản quyền với định dạng là:

*@ copyright Chủ Sở Hữu*

- @license: Địa chỉ tới giấy phép tác quyền với định dạng ví dụ:

*@license <http://www.example.com/path/to/license.txt> Tên Giấy Phép*

# DocBlock tags

- @var: Thông tin về kiểu dữ liệu và mô tả các biến hoặc properties của class. Định dạng:

*@var kiểu mô tả*

- @param: Thông tin về kiểu dữ liệu và mô tả các tham số của function hay. Định dạng:

*@param kiểu \$tênThamSố mô tả*

- @return: Thông tin về kiểu dữ liệu và mô tả giá trị trả về của function hay method. Định dạng:

*@return kiểu mô tả*

# DocBlocks

```
/**
 * A simple class
 *
 * This is the long description for this class,
 * which can span as many lines as needed. It is
 * not required, whereas the short description is
 * necessary.
 *
 * It can also span multiple paragraphs if the
 * description merits that much verbiage.
 *
 * @author Jason Lengstorf <jason.lengstorf@ennuidesign.com>
 * @copyright 2010 Ennui Design
 * @license http://www.php.net/license/3_01.txt PHP License 3.01
 */
```

# DocBlocks

```
class SimpleClass
{
    /**
     * A public variable
     *
     * @var string stores data for the class
     */
    public $foo;

    /**
     * Sets $foo to a new value upon class instantiation
     *
     * @param string $val a value required for the class
     * @return void
     */
    public function __construct($val) {
        $this->foo = $val;
    }
}
```

```
/**
 * Multiplies two integers
 *
 * Accepts a pair of integers and returns the
 * product of the two.
 *
 * @param int $bat a number to be multiplied
 * @param int $baz a number to be multiplied
 * @return int the product of the two parameters
 */
public function bar($bat, $baz) {
    return $bat * $baz;
}
}
```



# BÀI TẬP



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY





## Bài tập 2.1

Tạo class User gồm các properties:

- \$username
- \$password
- \$firstName
- \$lastName

## Bài tập 2.2

- Tạo constructor cho class:

\_\_construct(\$username, \$password, \$firstName, \$lastName)

*Lưu ý: password được hashed bằng hàm password\_hash("chuỗi cần hash", PASSWORD\_DEFAULT);*

- Tạo public method:
  - getFullname(): trả về \$firstName \$lastName
  - getUsername(): trả về \$username

## Bài tập 2.3

- Tạo file register.php, tạo form đăng ký thông tin gồm username, password, firstName, lastName.
- Xuất ra thông tin của user vừa nhập bên dưới form theo table:

| Username | Full name |
|----------|-----------|
|          |           |

## Bài tập 2.4

- Tạo public method login(\$username, \$password), kiểm tra nếu \$username = admin và \$password = giá trị hash của 12345 thì return true.

*Lưu ý: sử dụng hàm password\_verify ( string \$password , string \$hash ) để kiểm tra password nhập vào & password đã hashed.*

- Tạo file login.php chứa form login, nếu login thành công thì hiện thông báo “Logged in successfully” bên dưới form.

## Bài tập 2.5

- Tạo class Student kế thừa từ class User.
- Student có thêm property là \$gpa (điểm trung bình)
- Viết constructor cho Student kế thừa từ User nhưng thêm \$gpa
- Viết public method xếp loại cho Student rank() như sau:
  - $GPA < 5$ : yếu,  $5 \leq GPA < 7$ : trung bình,  $7 \leq GPA < 8$ : khá,  $GPA \geq 8$ : giỏi
  - Hàm return về xếp loại của một sinh viên.
- Tạo file student.php, tạo form nhập thông tin student và xuất ra như bảng bên dưới:

| Username | Full name | GPA | Rank |
|----------|-----------|-----|------|
|          |           |     |      |

# Thanks for your attention!



**FACULTY OF INFORMATION TECHNOLOGY**  
**Thu Duc College of Technology**

Phone: (+848) 22 158 642

Email: [fit@tdc.edu.vn](mailto:fit@tdc.edu.vn)

Website: [fit.tdc.edu.vn](http://fit.tdc.edu.vn)



FACULTY OF INFORMATION TECHNOLOGY  
THU DUC COLLEGE OF TECHNOLOGY

