MVC框架第1天

[学习目标 2](#_Toc522636544)

[MVC框架思想原理 2](#_Toc522636545)

[1、MVC概述 2](#_Toc522636546)

[2、MVC各组件的功能 2](#_Toc522636547)

[4、实例：MVC思想简单演示 3](#_Toc522636548)

[（1）需求分析 3](#_Toc522636549)

[（2）控制器文件controller.php 4](#_Toc522636550)

[（3）模型类文件model.class.php 5](#_Toc522636551)

[（4）视图文件view.html 5](#_Toc522636552)

[MVC实例：学生信息管理 6](#_Toc522636553)

[1、实例：使用MVC思想展示学生信息列表 6](#_Toc522636554)

[（1）学生控制器文件：StudentController.php 6](#_Toc522636555)

[（2）学生模型类文件：StudentModel.class.php 7](#_Toc522636556)

[（3）数据库工具类文件：Db.class.php 8](#_Toc522636557)

[（4）学生信息首页视图文件：StudentIndexView.html 9](#_Toc522636558)

[2、实例：使用MVC思想删除学生信息 10](#_Toc522636559)

[（1）学生信息首页视图文件：StudentIndexView.html 10](#_Toc522636560)

[（2）学生控制器文件：StudentController.php 11](#_Toc522636561)

[（3）学生模型类文件：StudentModel.class.php 12](#_Toc522636562)

[3、MVC总结 13](#_Toc522636563)

[MVC第1个版本：整合学生模块和新闻模块 13](#_Toc522636564)

[1、需求分析 13](#_Toc522636565)

[2、复制学生模块相关文件 14](#_Toc522636566)

[3、修改新闻模块相关文件内容 14](#_Toc522636567)

[4、每个视图文件添加简单导航栏 15](#_Toc522636568)

[5、添加首页文件index.php，并跳转到默认控制器文件 15](#_Toc522636569)

[MVC第2个版本：基础模型类的实现 15](#_Toc522636570)

[1、需求分析 15](#_Toc522636571)

[2、创建基础模型类文件：BaseModel.class.php 16](#_Toc522636572)

[3、其它模型类继承基础模型类 16](#_Toc522636573)

[MVC第3个版本：工厂模型类 17](#_Toc522636574)

[1、需求分析 17](#_Toc522636575)

[2、创建工厂模型类：FactoryModel.class.php 17](#_Toc522636576)

[3、修改控制器文件代码 18](#_Toc522636577)

[MVC第4个版本：控制器类的实现 18](#_Toc522636578)

[1、需求分析 18](#_Toc522636579)

[2、将过程代码提升为函数调用 19](#_Toc522636580)

[3、将函数调用提升为控制器类方法 20](#_Toc522636581)

[4、修改控制器类文件扩展名为.class.php 21](#_Toc522636582)

[5、修改视图文件导航栏链接地址 21](#_Toc522636583)

[MVC第5个版本：添加学生信息 21](#_Toc522636584)

[1、学生首页视图文件：StudentIndexView.html 21](#_Toc522636585)

[2、学生控制器添加add()方法 22](#_Toc522636586)

[3、学生添加视图文件：StudentAddView.html 22](#_Toc522636587)

[4、学生控制器添加insert()方法 23](#_Toc522636588)

[5、学生模型类添加insert()方法 23](#_Toc522636589)

[MVC第6个版本：控制器类的进一步优化 24](#_Toc522636590)

[MVC第7个版本：删除新闻数据 24](#_Toc522636591)

# 学习目标

* MVC框架思想原理
* MVC框架简单实例演示
* MVC第1个版本：整合学生模块和新闻模块
* MVC第2个版本：基础模型类的实现
* MVC第3个版本：模型类的单例工厂类
* MVC第4个版本：控制器类的实现
* MVC第5个版本：添加学生信息
* MVC第6个版本：控制器类的进一步优化
* MVC第7个版本：删除新闻数据

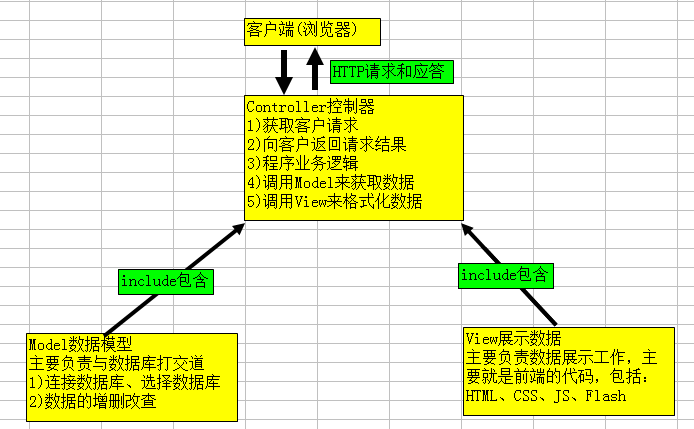
# MVC框架思想原理

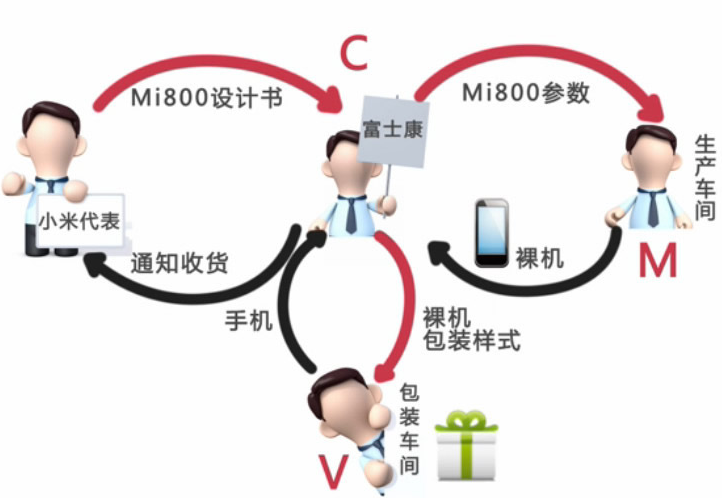
## 1、MVC概述

* MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写；
* MVC是一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码；
* MVC中，每个模块只做自己范围内的事情，不是自己的事情千万不做；
* 例如：开饭店，自己不生产食用油，而是找专门生产食用油的厂商；自己也不生产面粉，而是去找专门生产面粉的厂商；而面粉厂商，它也不种小麦，而去找种小麦的农民，……

## 2、MVC各组件的功能

* Model（数据模型）用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。
* View（视图）是用于处理数据显示的部分。通常视图是依据模型数据创建的。
* Controller（控制器）是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。



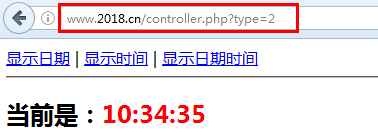


## 4、实例：MVC思想简单演示

### （1）需求分析

在网页上，单击不同的链接，显示不同的日期或时间信息：

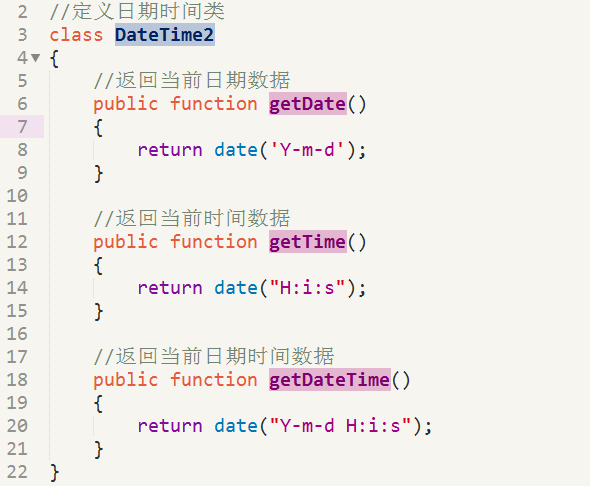
* 点击“显示日期”链接，则显示当前日期；
* 点击“显示时间”链接，则显示当前时间；
* 点击“显示日期时间”链接，则显示当前日期时间。



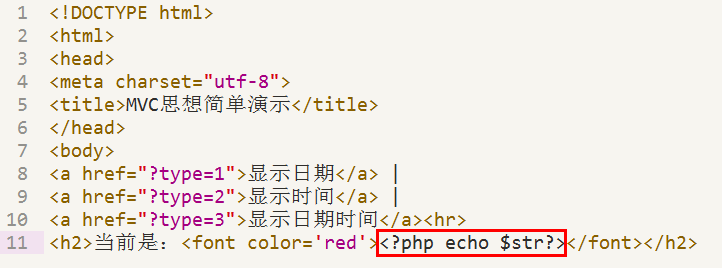
### （2）控制器文件controller.php



### （3）模型类文件model.class.php



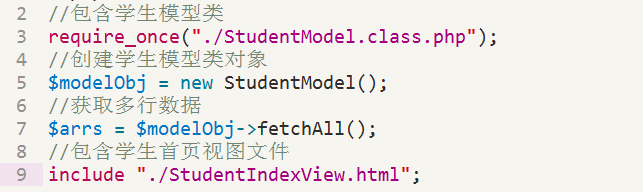
### （4）视图文件view.html



# MVC实例：学生信息管理

## 1、实例：使用MVC思想展示学生信息列表

### （1）学生控制器文件：StudentController.php



### （2）学生模型类文件：StudentModel.class.php



### （3）数据库工具类文件：Db.class.php



### （4）学生信息首页视图文件：StudentIndexView.html

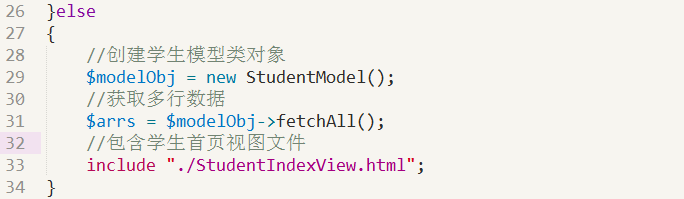


## 2、实例：使用MVC思想删除学生信息

### （1）学生信息首页视图文件：StudentIndexView.html



### （2）学生控制器文件：StudentController.php



### （3）学生模型类文件：StudentModel.class.php



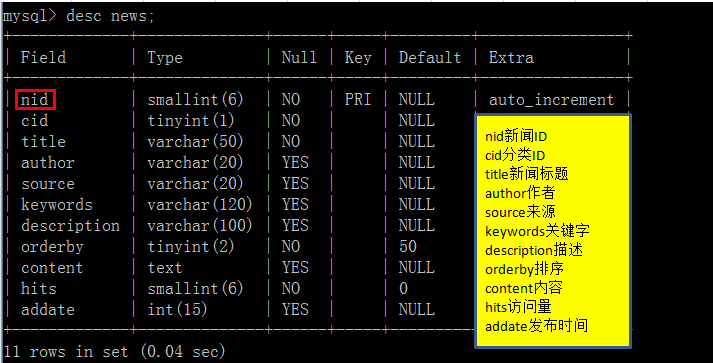
## 3、MVC总结

* 一个项目由多个功能模块构成：学生管理、新闻管理、用户管理、产品管理
* 一个功能只有一个控制器：StudentController、NewsController、UserController、ProductController
* 一个控制器对应一个模型类：StudentModel、NewsModel、UserModel、ProductModel
* 一个模型类对应一张数据表操作：student、news、user、product
* 一个控制器对应多个视图：StudentIndexView.html、StudentAddView.html、StudentEditView.html

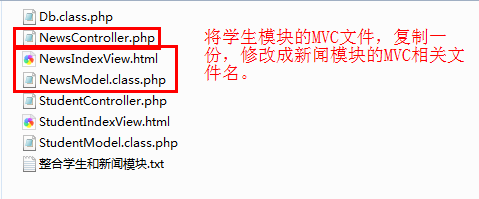
# MVC第1个版本：整合学生模块和新闻模块

## 1、需求分析

* 在MVC框架中，同时实现两个模块：学生信息模块、新闻模块；
* 将学生信息模块的MVC文件复制一份，并修改相应的文件名；
* 控制器文件：StudentController.php、NewsController.php
* 模型文类件：StudentModel.class.php、NewsModel.class.php
* 视图文件：StudentIndexView.php、NewsIndexView.php
* **提示：文件内部的相应路径，都要一一修改**。



## 2、复制学生模块相关文件



## 3、修改新闻模块相关文件内容



## 4、每个视图文件添加简单导航栏



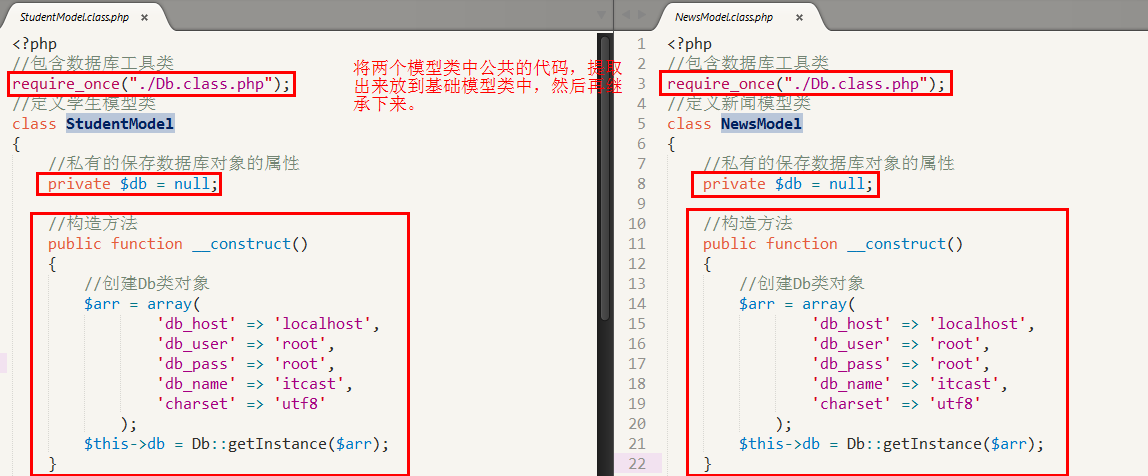
## 5、添加首页文件index.php，并跳转到默认控制器文件



# MVC第2个版本：基础模型类的实现

## 1、需求分析

* 我们将两个模型类中，公共的属性和方法，提炼出来，放入基础模型类(BaseModel.class.php)中。
* 其它模型类完成对基础模型类的继承。
* **重复的代码只写一遍，减少代码量，方便后期维护**。



## 2、创建基础模型类文件：BaseModel.class.php



## 3、其它模型类继承基础模型类



# MVC第3个版本：工厂模型类

## 1、需求分析

* 设计一个工厂模型类，用来生产不同模型类对象。
* 具体目标：设计一个类，该类可以"专门生产"模型类的对象。

## 2、创建工厂模型类：FactoryModel.class.php



## 3、修改控制器文件代码



# MVC第4个版本：控制器类的实现

## 1、需求分析

* 在MVC中，除了视图文件以外，几乎都是类文件；
* 将控制器过程代码，上升为控制器类代码。

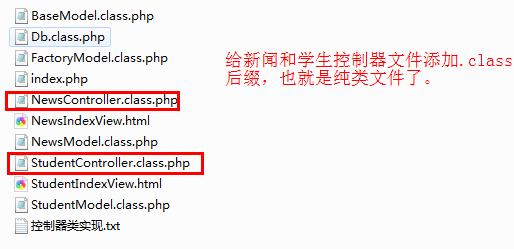
## 2、将过程代码提升为函数调用



## 3、将函数调用提升为控制器类方法



## 4、修改控制器类文件扩展名为.class.php

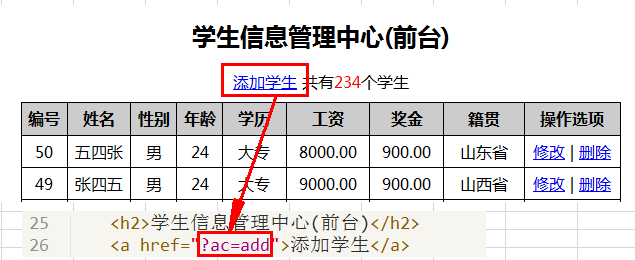


## 5、修改视图文件导航栏链接地址



# MVC第5个版本：添加学生信息

## 1、学生首页视图文件：StudentIndexView.html



## 2、学生控制器添加add()方法



## 3、学生添加视图文件：StudentAddView.html



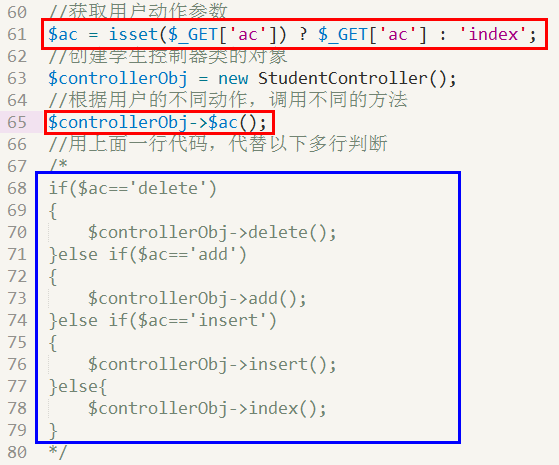
## 4、学生控制器添加insert()方法



## 5、学生模型类添加insert()方法



# MVC第6个版本：控制器类的进一步优化



# MVC第7个版本：删除新闻数据