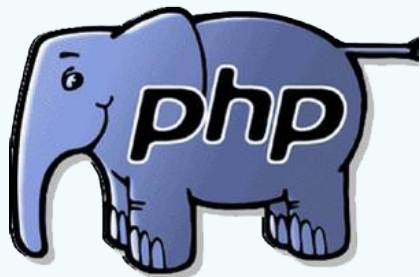




项目 10 ThinkPHP 框架 学生管理系统主页面制作



任务一

安装 ThinkPHP

任务二

输出 “hello,thinkphp”

任务三

制作首页面



1、PHP 发展史

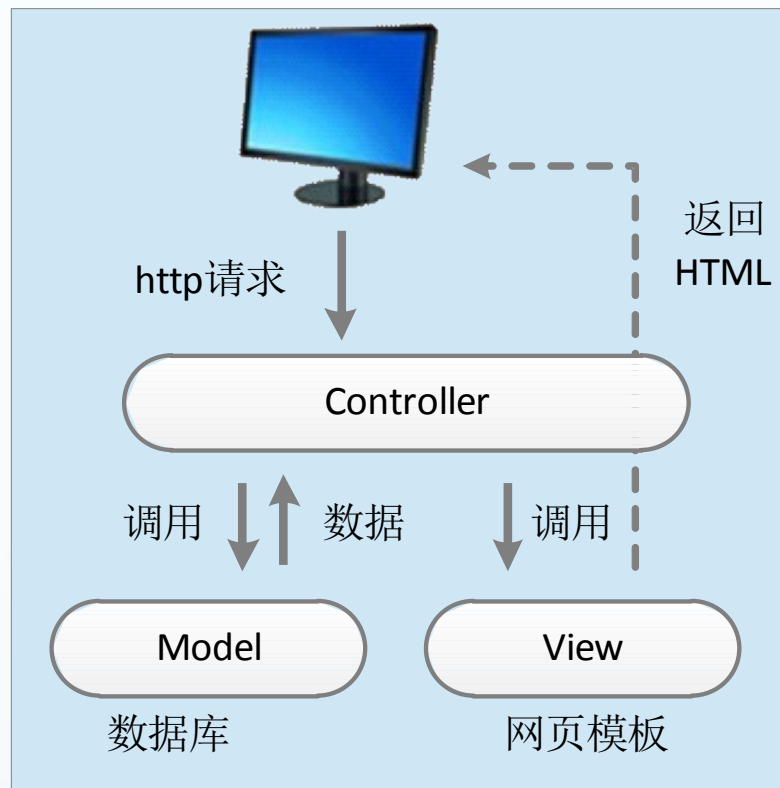
WEB 开发所经历的编程阶段

- 第一阶段：混合编程阶段
 - 优点：效率高
 - 缺点：不易于维护
- 第二阶段：模版引擎阶段 (**Smarty**)
 - 优点：前端与后端相分离
 - 缺点：效率较第一阶段低
- 第三阶段：**MVC** 框架设计阶段
 - 优点：维护方便
 - 缺点：效率比前 2 种阶段低，但可通过硬件的提升得到解决



2、MVC 框架

MVC 是一个框架模式，它强制性的使应用程序的输入、处理和输出分开。使用 MVC 应用程序被分成三个核心部件：**控制器（Controller）**、**模型（Model）**、**视图（View）**，它们各自处理自己的任务。



2、MVC 框架

(1) 控制器

- 控制器接受用户的输入并调用模型和视图去完成用户的需求，所以当单击 **Web** 页面中的超链接和发送 **HTML** 表单时，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型去处理请求，然后再确定用哪个视图来显示返回的数据。



2、MVC 框架

（2）模型

- 模型表示数据和业务规则。在 MVC 的三个部件中，模型拥有最多的处理任务。被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据，由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

（3）视图

- 视图是用户看到并与之交互的界面，模型获取的数据最终通过视图展现给用户。



2、MVC 框架

- 常见 MVC 框架

- ① **Zend Framework**：重量级框架，是 PHP 的官方框架，功能全面，但较臃肿，启动慢
- ② **YII**：重量级框架，简称“易框架”
- ③ **CodeIgniter**：轻量级框架，简称 CI 框架
- ④ **Laravel**：轻量级框架，使用人数众多
- ⑤ **ThinkPHP**：轻量级框架，中国人开发，相应的中文手册、中文网站，国内使用非常广泛，适合入门级的学习



3、ThinkPHP 目录结构

- 结合开发手册一起学习
- <https://www.kancloud.cn/manual/thinkphp/1689>

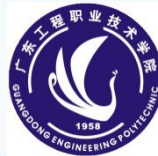
- ① 获取 Thinkphp
- ② 环境要求
- ③ 目录结构
- ④ 入口文件
- ⑤ 自动生成
- ⑥ 开发规范



3、ThinkPHP 目录结构

- ThinkPHP 框架的核心文件都在 ThinkPHP 下。
- \ThinkPHP\Conf 目录是 ThinkPHP 的**核心配置目录**，其中包含了 ThinkPHP 惯例配置文件，该文件中包含数据库连接信息、ThinkPHP 默认设定、URL 访问模式等默认配置。
- \ThinkPHP\Library\Think 是核心 ThinkPHP 类库包目录，其中包含了 App.class.php（应用程序类）、Controller.class.php（控制器基类）、Model.class.php（模型基类）、View.class.php（视图基类）等 ThinkPHP 运行所需的基础类文件。





3、ThinkPHP 目录结构

Application 目录结构

文件路径	文件描述
\Application\Common	应用公共模块
\Application\Common\Common	应用公共函数目录，为 Application 目录下的所有模块提供公共函数
\Application\Common\Conf	应用公共配置文件目录，为 Application 目录下的所有模块提供公共配置
\Application\Home	ThinkPHP 框架默认生成的 Home 模块
\Application\Home\Conf	模块配置文件目录，为 Home 模块提供配置信息
\Application\Home\Common	模块函数公共目录，为 Home 模块提供公共函数
\Application\Home\Controller	模块控制器目录
\Application\Home\Model	模块模型目录
\Application\Home\View	模块视图目录
\Application\Runtime	运行时目录
\Application\Runtime\Cache	模板缓存目录
\Application\Runtime\Data	数据目录
\Application\Runtime\Logs	日志目录
\Application\Runtime\Temp	缓存目录



案例展示

安装 THINKPHP3.2.3 ，结合开发手册查看相应的目录结构。

4、ThinkPHP 基本使用

详见开发手册【架构】 - 【模块化设计】

(1) ThinkPHP 新版采用模块化的架构设计思想，一个完整的 ThinkPHP 应用基于**模块 / 控制器 / 操作**设计

名称	描述
应用	基于同一个入口文件访问的项目我们称之为一个应用。
模块	一个应用下面可以包含多个模块，每个模块在应用目录下面都是一个独立的子目录。
控制器	每个模块可以包含多个控制器，一个控制器通常体现为一个控制器类。
操作	每个控制器类可以包含多个操作方法，也可能是绑定的某个操作类，每个操作是URL访问的最小单元。

模块化设计的思想中模块是最重要的部分，模块其实是一个包含配置文件、函数文件和 MVC 文件（目录）的集合。



4、ThinkPHP 基本使用

详见开发手册【基础】 - 【模块】

(2) 下面是一个应用目录下面的模块目录结构

```
Application 默认应用目录（可以设置）  
├─Common 公共模块（不能直接访问）  
├─Home 前台模块  
├─Admin 后台模块  
├─... 其他更多模块  
├─Runtime 默认运行时目录（可以设置）
```



4、ThinkPHP 基本使用

详见开发手册【基础】 - 【控制器】

（3）控制器：实质就是一个类文件

- 控制器类的命名方式是：控制器名（驼峰法，首字母大写）+**Controller**
- 控制器文件的命名方式是：类名+**class.php**（类文件后缀）



4、ThinkPHP 基本使用

详见开发手册【基础】 - 【控制器】

（3）控制器：实质就是一个类文件

• 控制器文件的开头代码：

- **namespace Home\Controller;**// 命名空间定义。这是系统的规范要求，表示当前类是 Home 模块下的控制器类，命名空间和实际的控制器文件所在的路径是一致的。
- **Use Think\Controller;** // 表示引入 Think\Controller 类库便于直接使用。
- **class IndexController extends Controller{}** // 定义类的具体代码

注意：命名空间定义必须写在所有的 PHP 代码之前声明，而且之前不能有任何输出，否则会出错



4、ThinkPHP 基本使用

详见开发手册【控制器】 - 【控制器定义】

（4）操作：一般来说，ThinkPHP 的控制器是一个类，而操作则是控制器类的一个公共方法。

```
<?php
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function hello(){
        echo 'hello,thinkphp!';
    }
}
```

定义了一个 Index 的控制类，该类含有一个操作 hello



4、ThinkPHP 基本使用

详见开发手册【架构】 - 【模块化设计】

(5) URL 访问:

一个典型的URL访问规则是 (我们以默认的PATHINFO模式为例说明, 当然也可以支持普通的URL模式) :

1. `http://serverName/index.php (或者其他应用入口文件) /模块/控制器/操作/[参数名/参数值...]`

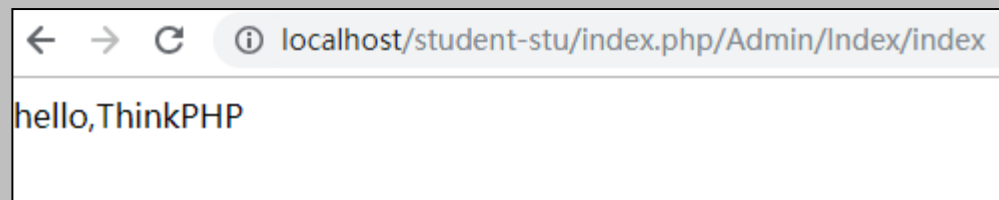
hello,thinkphp!





案例展示

页面输出





案例 展示

Index 页面

服务器信息列表

系统版本	Apache/2.4.9
apache版本	(Win64)
PHP版本	PHP/5.5.12
脚本地址	D:/wamp/www/student_stu/index.php
系统时间	2018-05-23

4、ThinkPHP 基本使用

- PHP 语法手册（菜鸟教程）
- <https://www.runoob.com/php/php-tutorial.html>
- ① `$_SERVER`
- ② `$_SERVER['SERVER_SOFTWARE']`
- ③ PHP 函数 `explode()`：把字符串打散为数组
- 例如：`$sys_info= explode(" ", $_SERVER[SERVER_SOFTWARE]);`
`dump($sys_info)`

```
array (size=3)
  0 => string 'Apache/2.4.9' (length=12)
  1 => string '(Win64)' (length=7)
  2 => string 'PHP/5.5.12' (length=10)
```



4、ThinkPHP 基本使用

详见开发手册【视图】 - 【模板赋值】

(6) assign()

- ThinkPHP 框架默认开启了模板引擎，在开启模板引擎的情况下，变量都需要使用 `assign()` 方法将变量分配给视图文件。示例代码如下：
- `assign()` 方法的第一个参数表示数据在视图中的名称，第二个参数表示变量名。可以使用 `$this->assign('name',$name);` 的方式，示例代码如下：
- 需要注意的是，`assign()` 方法必须在 `display()` 方法前调用，分配的变量数据才能显示到视图中。

```
$this->name=$name;
```



4、ThinkPHP 基本使用

详见开发手册【视图】 - 【模板渲染】

(7) display ()

- ThinkPHP 使用 display() 方法来显示视图， display() 方法有三种形式：

用法	示例	描述
不带任何参数	<code>\$this->display()</code>	系统会自动定位当前操作的模板文件
[模块 @][控制器 :][操作]	<code>\$this->display("Admin@Index:index")</code>	表示会输出 Admin 模块下的 View 目录下的 Index 目录下的 index.html 文件
完整的模板文件名	<code>\$this->display("../Temp/Public/index.html")</code>	表示输出项目根目录下的 Temp 目录下的 Public 目录下的 index.html 文件，注意使用这种方式一定要加上视图文件后缀。



详见开发手册【视图】 - 【模板定义】

(8) 视图的创建

- ThinkPHP 对模板文件进行目录划分，默认的模板文件定义规则是：
 - 视图目录 (View 目录) / 控制器名 / 操作名 + 模板后缀 (.html)
 - 如：User 控制器的 add 操作 对应的模板文件就应该是： ./Application/Home/View /User/add.html



4、ThinkPHP 基本使用

详见开发手册【模板】 - 【变量输出】

(9) 显示变量

- 在使用 **assign()** 方法为视图变量赋值后，就可以在视图文件中输出变量了。输出变量可以使用 **PHP** 原生语法，例如：

- 类似 Smarty 的模板引擎技术 Think Template，该模板引擎输出变量的语法如下：

```
{ $name }
```



4、ThinkPHP 基本使用

(9) 显示变量

- **ThinkTemplate** 模板标签默认的开始标记是 “{”，结束标记是 “}”。当然也可以通过配置文件对其进行修改，例如：

- 此时变量输出
- ```
'TMPL_L_DELIM'=>'<{'
'TMPL_R_DELIM'=>'}>',
```

```
<{$name}>
```

