Thinkphp总结

第一节：ThinkPHP 3.1.2 介绍及安装

1. tp是免费的开源的基于mvc模式的轻量级的PHP框架程序；
2. 学习TP必须学习面向对象；

3.框架就像是一个毛坯房 ，让我们去装修；

4.MVC：分为

M层 （model层 叫做模型，主要是负责数据操作，比如向数据库插入数据 查询数据等）

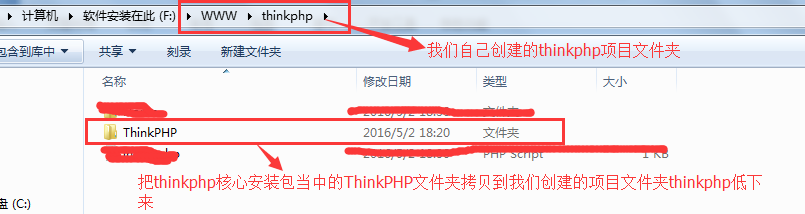
C层（控制器 ；叫做模块，用来描述项目的具体功能，比如订单模块，商品模块，描述主要的功能）

V层（视图层（模板），主要是显示前台页面的）

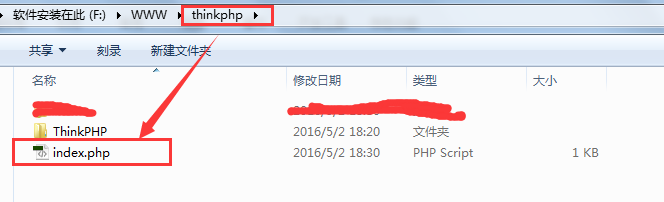
1. Thinkphp3.0以上要求的php版本必须是5.2.0以上版本！
2. 搭建一个项目：

A:首先在www目录下创建一个项目文件夹叫做thinkphp文件夹；

B:把‘ThinkPHP3.1.2\_核心包’当中的 “ThinkPHP”文件夹拷贝到我们创建的thinkphp文件夹下边来；



C:我们就要在项目的根目录（thinkphp目录）底下创建前台主入口文件index.php;



D:操作前台主入口文件index.php:



（给个提示：确定应用路径./home/ 必须要在home的后边加上/ 必须得加 严格要求！！！）

E:访问我们的项目http://localhost/thinkphp/会自动访问前台主入口文件index.php文件；就会出现欢迎界面：并且在项目的根目录自动创建一个Home的文件夹出来；将来前台的所有代码都会放置在Home这个文件夹下边；



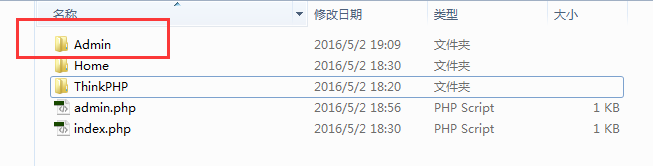
就这个样子前天创建出来了 接下来我们创建后台：

F:复制一份index.php改名字为admin.php 这是后台的主入口文件；

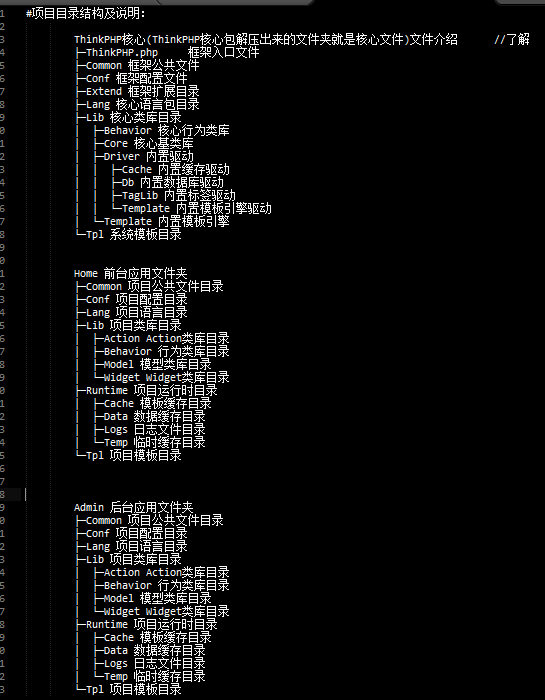
G：操作后台主入口文件：



H:访问后台入住口文件：<http://localhost/thinkphp/admin.php那么就会自动创建Admin后天文件夹；>将来后天的所有代码都会放置在这个Admin目录下边；



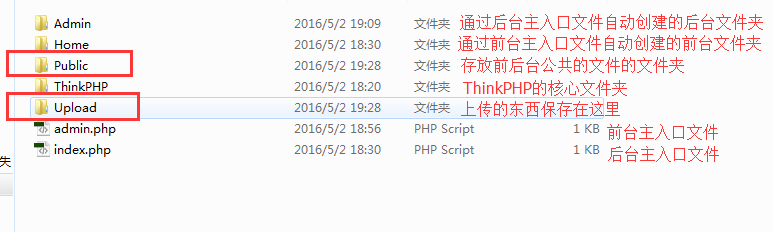
目录的说明：



补充知识：

前台和后台公共的一些文件我们可以创建文件夹Public来存放这些公共的文件；

上传的文件我们需要创建Upload文件夹来存放；



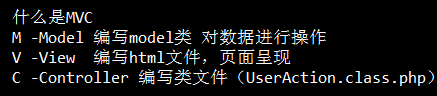
1. ：ThinkPHP.3.1.2.MVC模式和URL访问
2. MVC当中我们在使用

M层的时候一般是要编写一个类文件也叫作Model类来操作要操作的数据；

C 层控制器也是要编写类文件的，比如商品模块就要编写一个商品的控制器类文件；也是负责某个功能的实现的类文件；

V 负责前台界面的呈现就是来编写模板文件的；

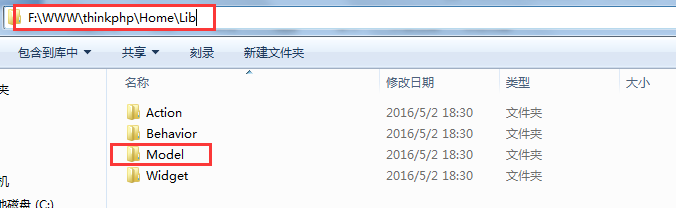
如图所示：

.

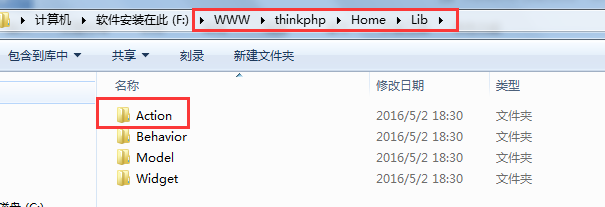
1. ThinkPHP非常的灵活，有时候没有M或者没有C都能正常的工作；就看这么使用怎么写；
2. MVC分别对应的目录：

V（视图层） 对应的目录是项目下的前台目录Home/TPL目录就是存放前台模板的目录；

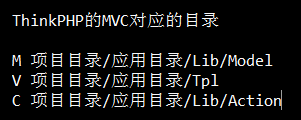
M（model层） 对应的目录是项目下的前台目录Home/lib/model/目录



C（控制器） 对应的目录是项目下的前台目录Home/lib/action/目录



如下图：

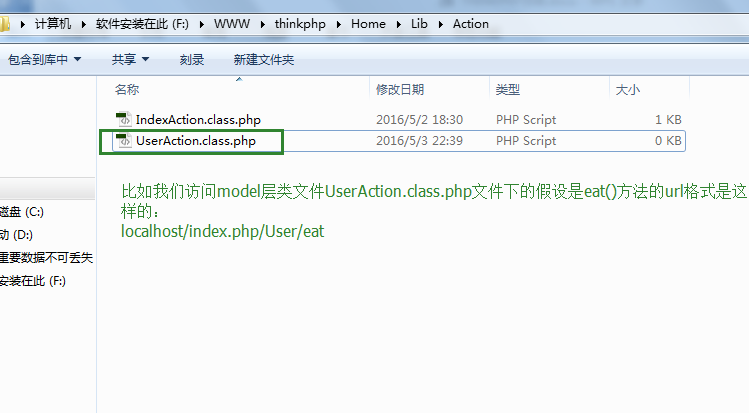
（应用目录就是Home文件夹）

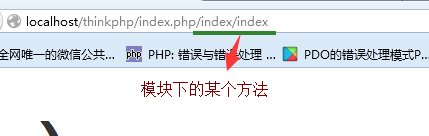
1. url 访问c层（控制器）



(注意：在命名C层类文件的时候前边的名字我们可以随便起 但是必须加上 Action.class.php才行： （自定义名称Action.class.php） 控制器C层的类文件是这么命名的哦！！！)

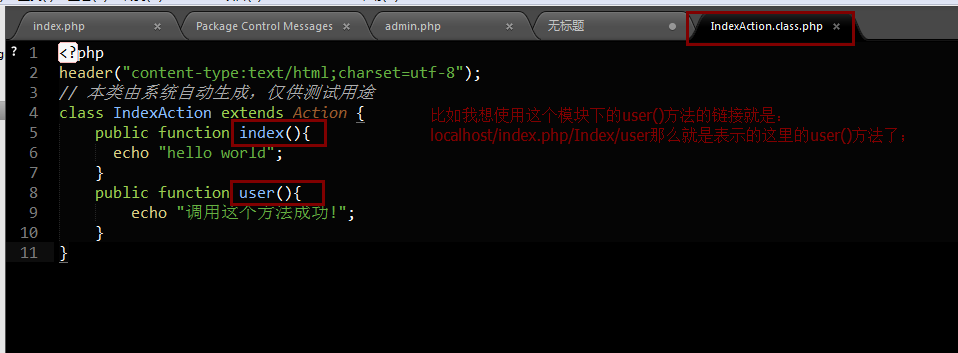
当然C层控制前类文件当中类的名称也是要和类文件名称一样：IndexAction 如果是UserAction.class.php的C层类文件 那么类名称就是UserAction;





当我们访问localhost/index.php前台主入口文件的时候默认的就是去C层控制器当中IndexAction.class.php文件当中的index()方法；即：localhost/index.php/Index/index方法；

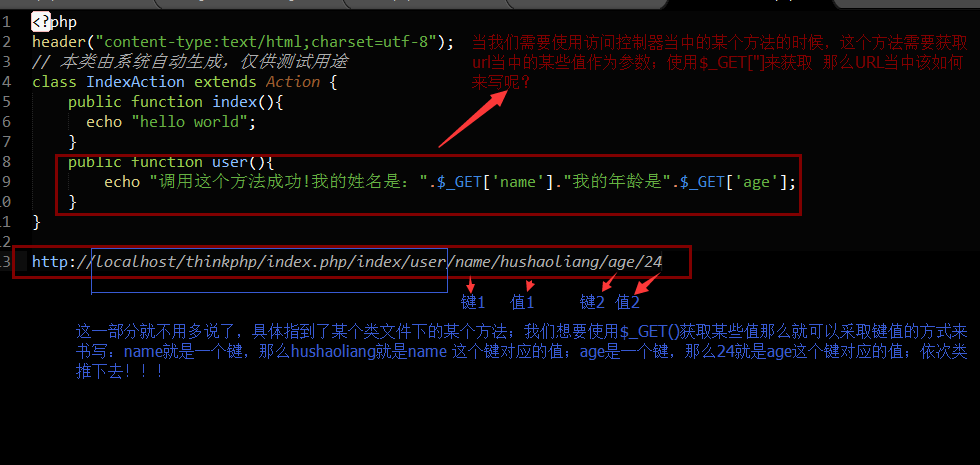
什么叫做模块？比如我要开发商品功能模块那么我们就要创建出一个GoodsAction.class.php类文件，封装起来就是一个商品功能的控制前文件，这就是一个模块！这是一个规则呀准守



Url的四种访问方法：

1.PathInfo 模式：

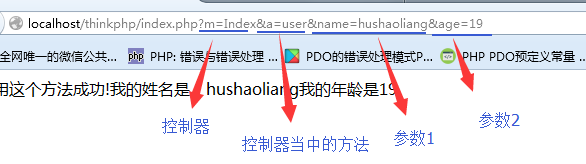
<http://域名/项目名/入口文件/模块名/方法名/键1/值1/键2/值2>



2普通模式（最常见 但是在TP当中不推荐使用的模式）

<http://域名/项目名/入口文件?m=模块名&a=方法名&键1=值1&键2=值2>

(这是在TP当中，所以要用m来代表模块c层控制器Index 用a 来代表控制器类下的方法；键值对的形式表示参数；

)

1. rewirite模式（了解）
2. 兼容模式（了解） 详情去看视频

如何修改url上的分割符号：

正常情况下我们都是使用/作为url的分割符号：localhost/index.php/Index/user/name/hushaoliang

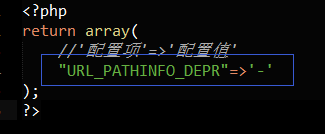
但是我们想要改变这种分割符号改成-

Localhost/index.php/Index-user-name-hushaoliang

(前边的locahost/index.php/Index 这些的分隔符号是不能改变的，也就是说改变的只是指定了某个控制器/Index之后的才能改变比如/Index-user这个Index控制器下的就可以使用-了；)

首先我们得找到应用文件夹（也就是前台文件夹Home/config/config.php 框架配置文件） 里面是renturn回来一个数组 需要我们去编写配置选项：

“URL\_PATHINFO\_DEPR”=>”-” 就行；如图：



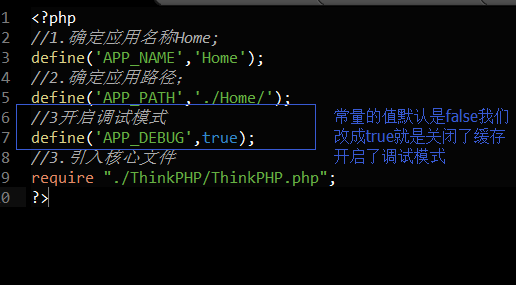
当我们改变了配置选项之后不会起作用那是因为缓存的问题；在Home/Runtime/下放的是缓存文件，这个Runtime可以时不时的删除掉，因为这是自动生成的；存放的都是缓存文件；把它删除很多东西就不会再走缓存了，也就起效果了；

如何在我们的开发过程当中关闭缓存？

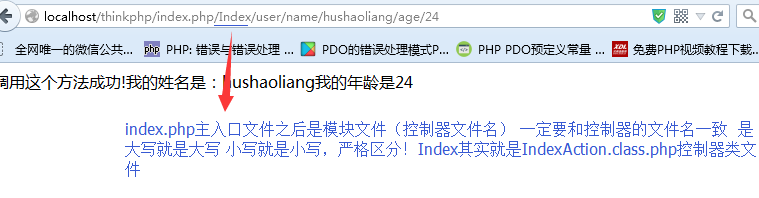
因为在开发过程当中如果开启缓存的话那么很多的修改不会立马见效 耽误事！所以我们在开发的过程当中就要关闭缓存的功能；如何关闭呢？

我们需要在前天主入口文件当中写一个常量来开启调试模式那么就自动的关闭了缓存开启了调试模式：

Define(“APP\_DEBUG”,true);

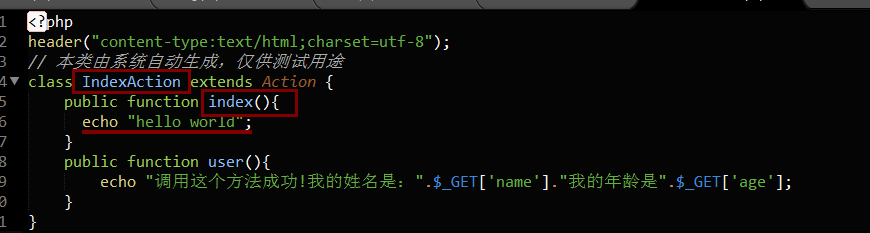


还需要注意的一点就是：



1. ：ThinkPHP 3.1.2 输出和模型使用
2. TP的输出有两种方法：

A:就是我们平时讲的这种最原始的PHP输出方法，我们通过url 访问一个控制器比如是Index控制器当中的index方法；然后在index方法当中我们输出了想要的内容，这是一种输出：



B:就是我们最常用的方法：

我们在控制器文件当中不直接输出 而是指向某个.html文件页面；使用的是$this->display();

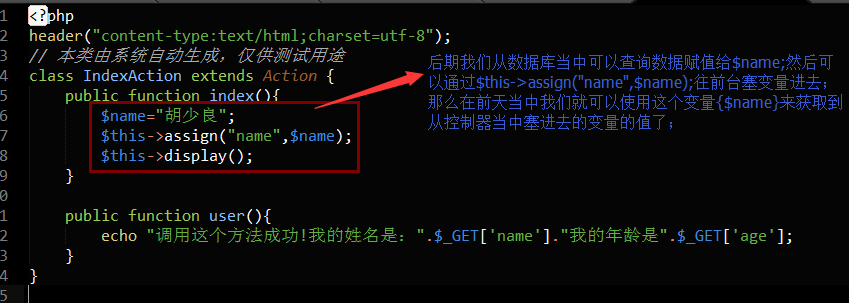


我们使用$this->display()的作用有两个：首先要在V层（也就是./Home/Tpl/下边）建立一个同访问的模块控制器方法一样的文件夹也即是Index控制器，那么就建立一个Index文件夹；在这个文件夹当中再建立一个同访问的控制器下的方法同名的html文件，也就是index.html文件，然后在index.html文件当中来编写想要输出的内容就行了；

（千万要记住：使用了MVC当中的V层也就是视图层，那么我们就要在V层文件夹./Home/Tpl/文件夹下边建立一个同访问的控制器文件相同名称的文件夹，假设访问的是userAction.class.php那么我们就得在./Home/TPl/文件夹下边建立一个User的文件夹 访问的是UserAction.class.php控制器当中的user()方法，那么我们就得在刚建立的User文件夹下边建立一个user.html文件来写入想要输出的内容； 控制器名对应文件夹名，方法名对应文件名！！！）

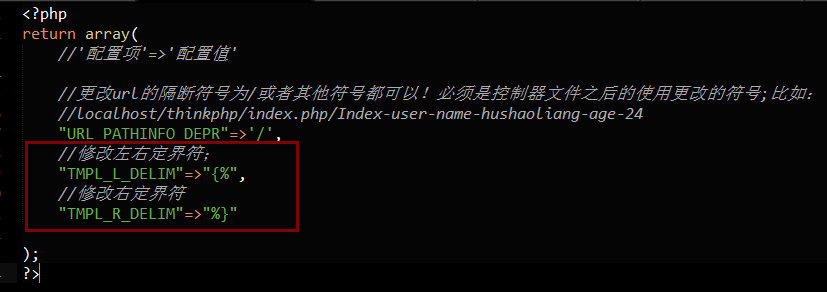
1. TP当中是如何让前台来显示动态数据的呢？

我们通过C层控制器指定了要显示的模板，使用的$this->display() 那么如果我们想要让前台显示动态数据那么也得要在C层控制器当中来进行操作，其实和smarty当中的差不多，都是往前台塞变量；使用的是$this->assign();详细见图解：



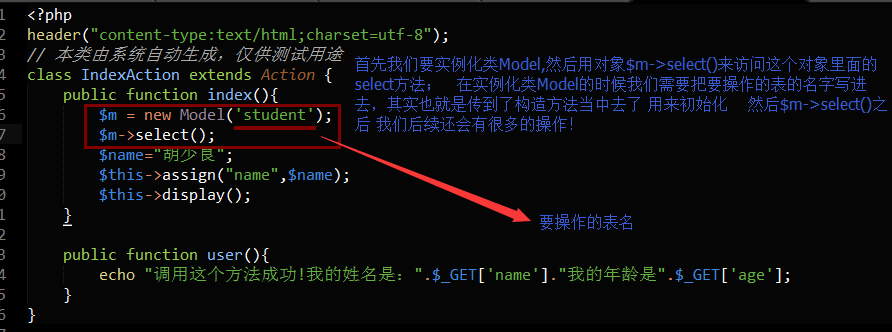
1. 如何来修改左右定界符：

在TP当中我们默认的在前台使用的定界符是{}但是不建议使用，原因和在smarty当中的原因是一样的，如果签单有JS出现就会乱掉；所以我们需要修改左右定界符号：  
我们还是要像修改url分割符那样的去找到前台的配置文件当然就是在./Home/conf/config文件；修改的地方如下：



1. 如何在控制器当中获取到数据库当中的一些值：

详情参考下图：（我们还没有配置TP链接数据库的一些配置信息所以下边案例只是参考）（建议在实例化Model类的时候里面的表明首字母大写Student）



1. 上边我们讲到还没有配置TP链接数据库的基本信息，那么该如何配置呢？

其实就是在前台的配置文件当中来进行配置：./Home/config/config.php文件当中进行配置链接数据库的选项：

'DB\_TYPE'=>'mysql', //设置数据库类型

'DB\_HOST'=>'localhost',//设置主机

'DB\_NAME'=>'thinkphp',//设置数据库名

'DB\_USER'=>'root', //设置用户名

'DB\_PWD'=>'', //设置密码

'DB\_PORT'=>'3306', //设置端口号

'DB\_PREFIX'=>'tp\_', //设置表前缀



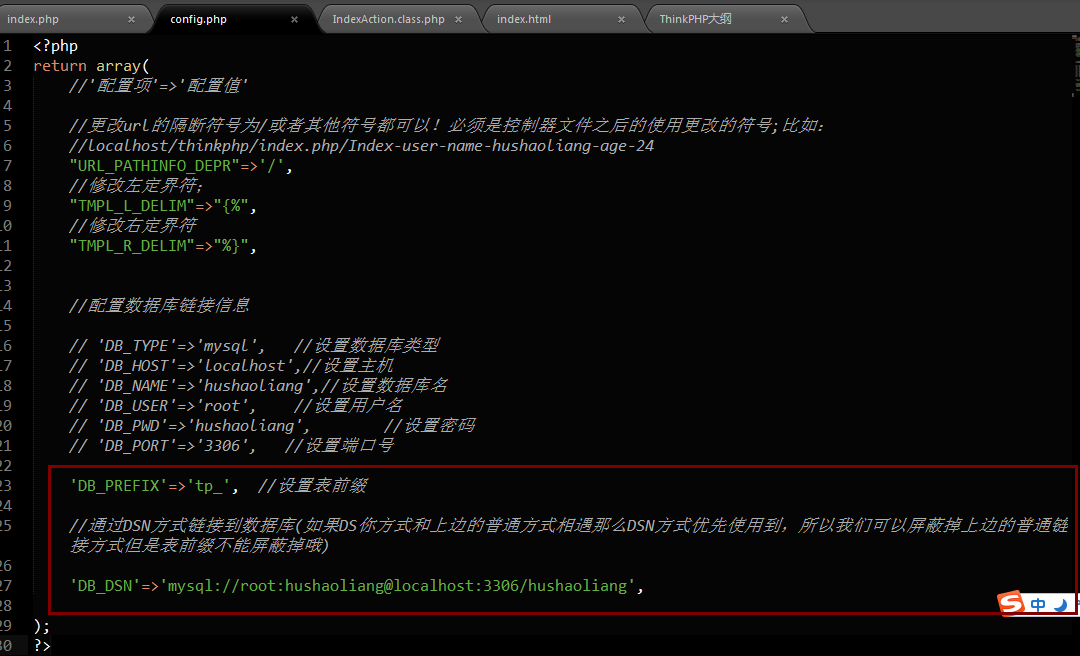
6.到现在我们就学会了TP的中心思想：

我们在地址栏通过url的方式访问不同C层控制器模块下的方法；我们通过C层控制器来进行功能的操作，在c层利用Model类对数据库进行操作，然后通过$this->assign()以及$this->display()来指定模板和往前台塞变量，然后通过V层来进行显示 就是这么一个思想；

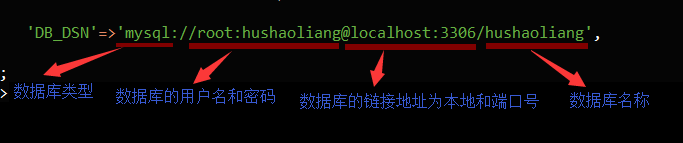
第四节：ThinkPHP.3.1.2.输出和模型使用

7.除了上边的了配置链接数据库信息的普通方式之外，还有一种方式叫做DSN方式：

'DB\_DSN'=>'mysql://root:hushaoliang@localhost:3306/hushaoliang',



8.当DSN方式和普通的方式相遇的时候那么优先选择DSN方式来配置数据库选项；



1. **我们在控制器也就是c层需要使用到Model层来访问数据库的一些信息；之前我们是这么用的:**

**用的是new Model（）实例化一个Model类; 把实例化了的$m叫做模型化实例；**

**但是我们有简单的方法就是大M方法（括号里面的表名字首字母必须大写）：**



**这里的大M是TP提供给我们的简单的写法，完全可以代替new Model（）来使用；$m叫做实例化模型；**

1. **我们实例化了Model类或者说使用了大M那么就会生成一个实例化的模型$m;那么我们就可以使用实例化了的模型$m对数据库进行增删改查的操作；ThinkPHP当中我们把增删改查叫做c u r d;**

**单词 对应方法**

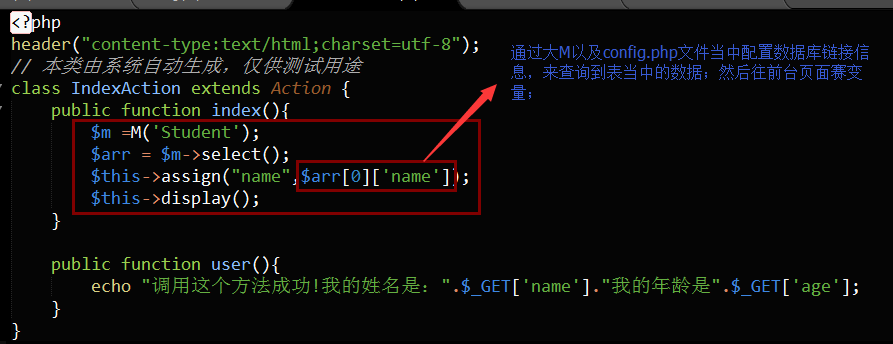
**增 -C Create $m->add()**

**删 -D Delete $m->delete()**

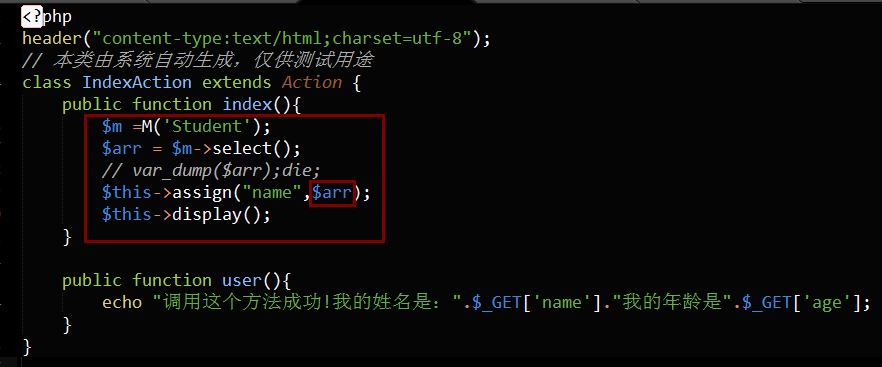
**改 -U Update $m->save()**

**查 -R Read $m->select() （前边用过哦）**

**11.（是这样往前台塞变量的哦！）**

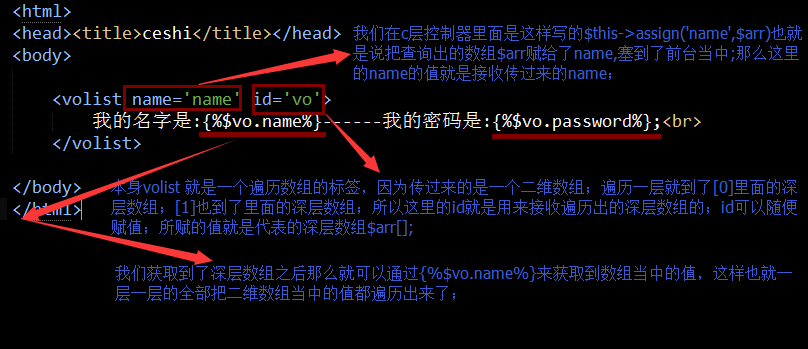


1. 不仅可以往前台页面当中塞单个的变量，也可以往前台页面当中塞一个数组进去：



但是往前台当中塞进去了数组如何在前台页面当中遍历出来呢？

我们要使用到TP当中的<volist name=’’id=’’></volist>标签：



1. 我们开启调试功能当中的page\_trace

A:首先要开启调试功能（在前台主入口文件当中设置，前边设置过可以屏蔽掉缓存哦）：

define('APP\_DEBUG',true);

B:在配置文件config.php文件当中开启trace:

'SHOW\_PAGE\_TRACE'=>true,//开启页面Trace

开启了这么那么就会在页面当中的右下角出现一个小图标：



点开之后会出现很多的项，错误，文件等等；

1. ：5.ThinkPHP 3.1.2 CURD特性
2. Model层对数据的操作无非就是增删改查，在TP当中我们叫做c u r d ;
3. 对数据的读取：

假设c层控制器当中是这样的：

$m=new Model('User');

$m=M('User');

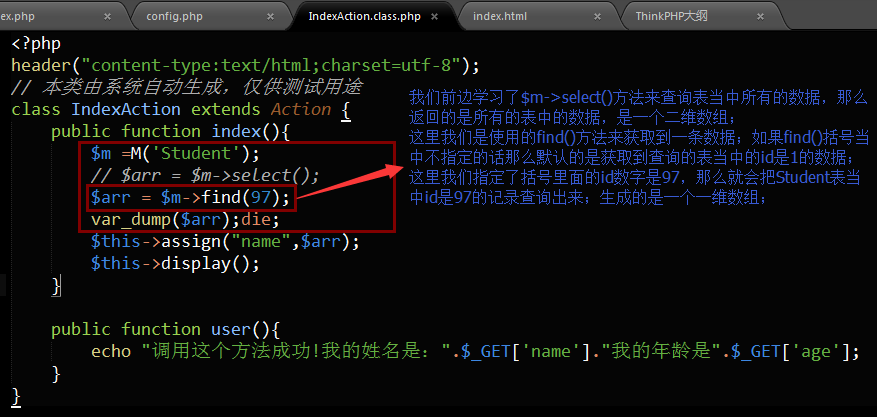
A：：： 我们使用select来查询读取数据：

select

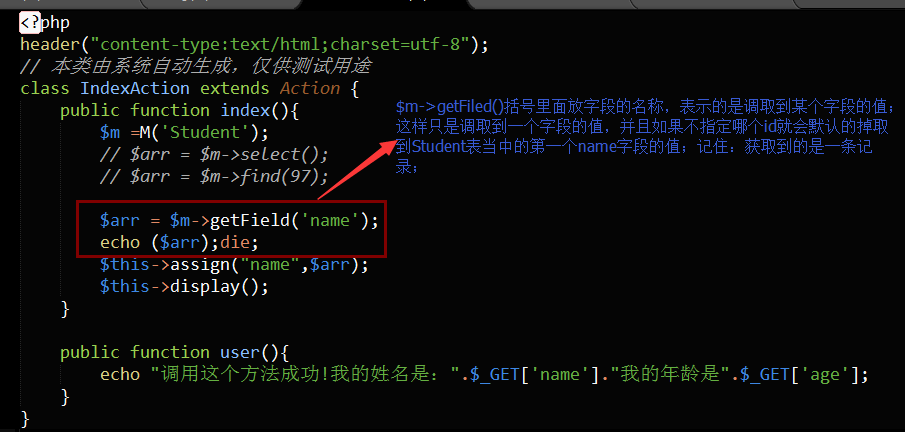
$m->select();//获取所有数据，以数组形式返回



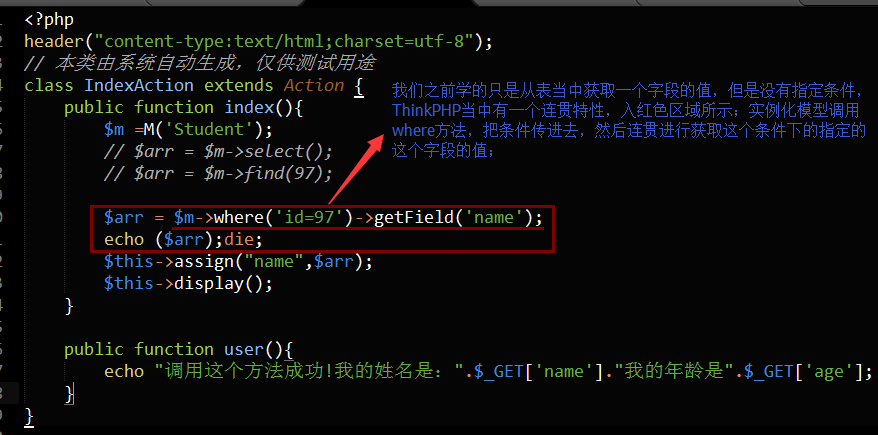
B:::除了可以使用select我们还能使用find来查询数据：



C:::除了上述的两种方法之外还有另一种查询的方式就是$m->getField() 括号里面放字段名称；

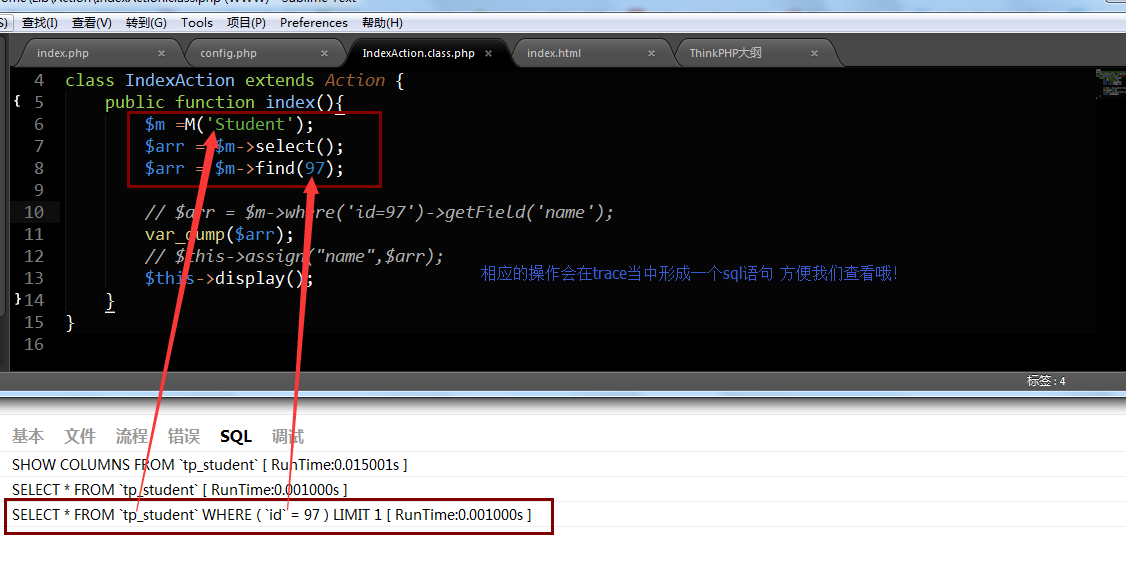


那么如何才能够掉取到某个id下的某个字段的值呢？



在学习$m->getField()的时候我们发现页面当中的trace消失了；原因是我们没有让页面显示也就是说没有让程序走到$this->display( ) 这一步！

当我们使用了$m->select() 或者$m->getField()等之后我们可以查看页面trace观察的到里面的sql语句的情况：



大家发现了没有：

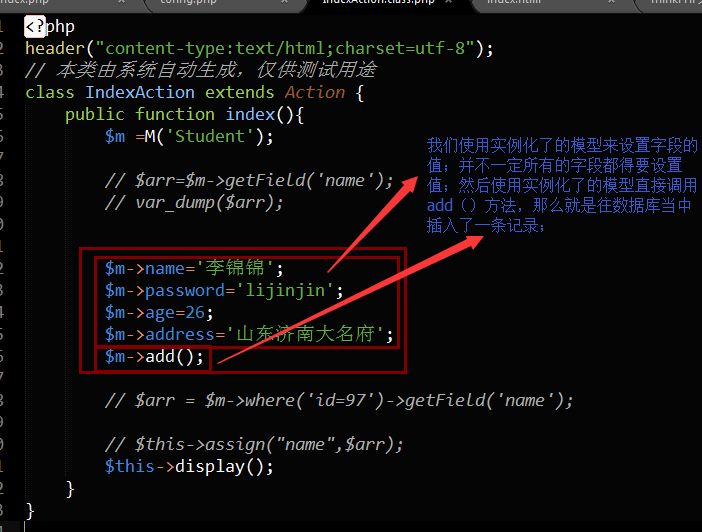
$m->select()获取整张表的数据

$m->find(97)指定某个id的值获取的是一条记录

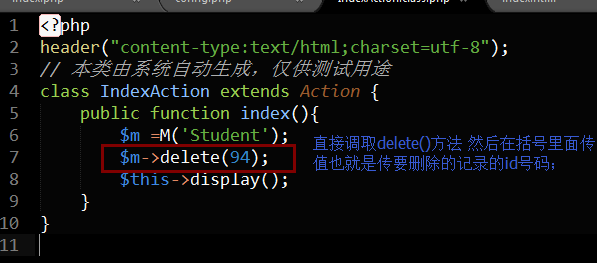
$m->getFiled()获取某个字段的值；

这样想获取什么就用什么，想当灵活了！

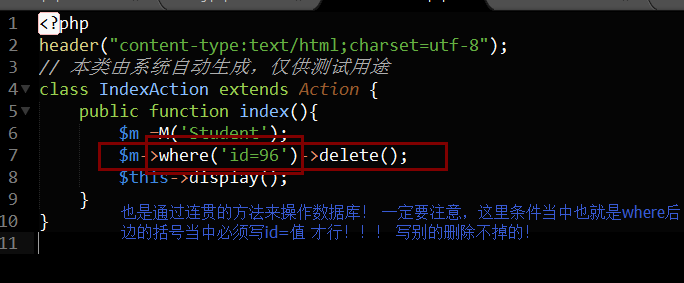
1. 对数据的添加：



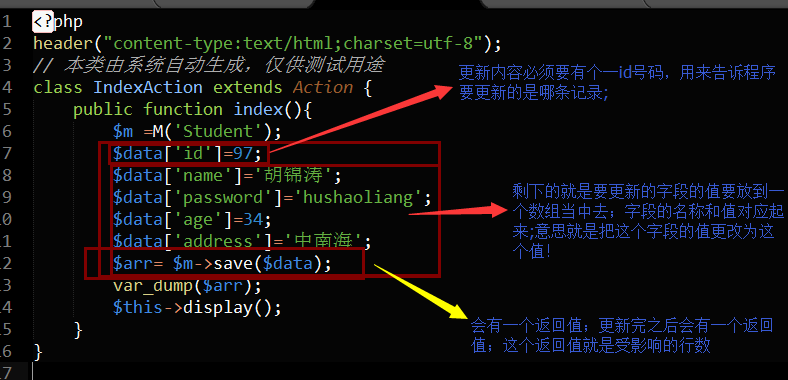
4.对数据的删除；（delete后边的括号当中只能放被删除的记录的id号码！）



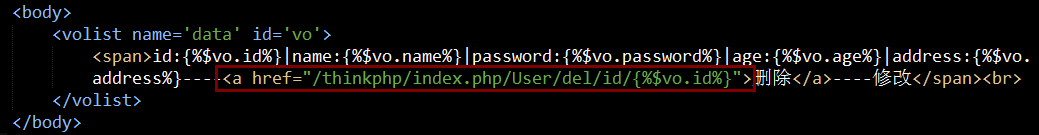
我们可不可以也像$m->getField()那么利用where来传一些条件进去呢？



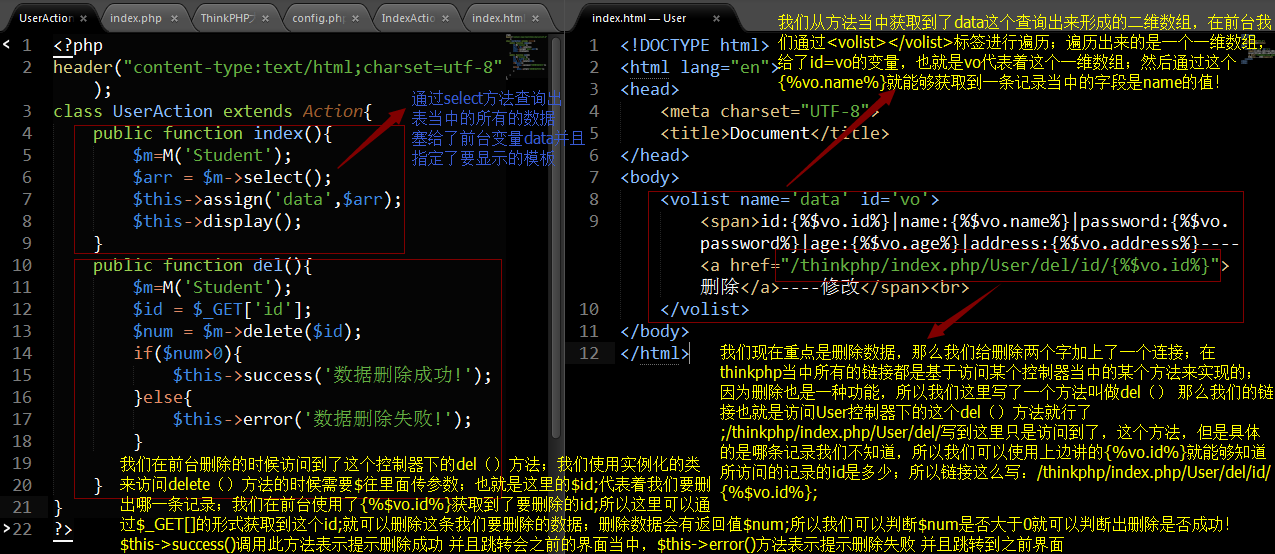
1. 更新数据：



1. 删除数据也是有返回值的；返回值是受影响的行数；
2. 更改数据也有返回值；返回值是受影响的行数；
3. 插入一条数据也是有返回值的；返回值是返回来的插入这条数据的id号码；
4. 读取数据当然就是返回回来的那些查询出来的数据了；
5. ：ThinkPHP.3.1.2.CURD演示.1
6. 链接地址的问题（参考2）

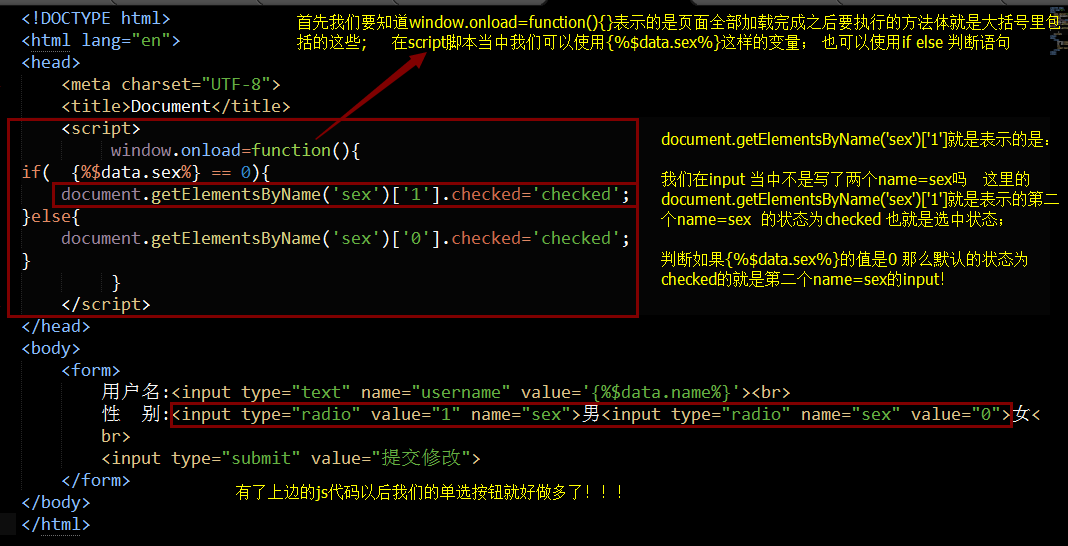


1. 在ThinkPHP当中所有的链接都是基于访问c层控制器来进行的；不管是在任何地方都得要通过访问控制器然后在所访问的控制器当中指定要显示的html模板或者说指定要跳转的地址；
2. 在thinkphp当中我们是如何在前端界面当中删除一条数据的：



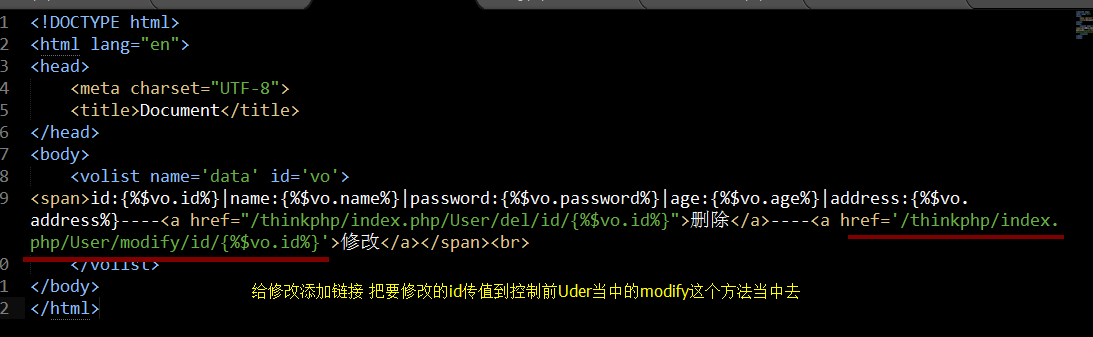
$this->success()和$this->error() 看上图！

4.在thinkphp当中当我们修改的时候默认选中的就是当时我们所选择的那个单选按钮：



1. ThinkPHP当中如何修改表单：

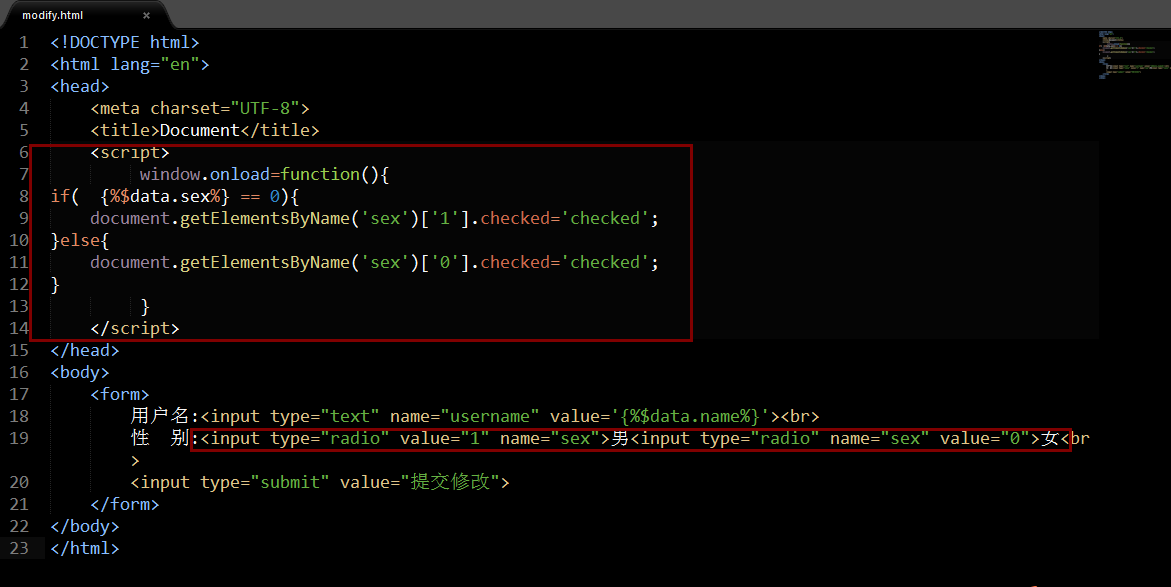
A:给修改添加链接/thinkphp/index.php/modify/id/{%$vo.id%}



B.当我们点击修改的时候通过User控制器当中的modify方法来访问到呈现出来的修改界面modify.html(我们利用传递过来的id去查询到要修改的对应的这条记录，然后赛变量到前台页面modify.html当中去；把我们之前已经添加好的数据显示出来，方便我们看到以前的数据好来修改；)



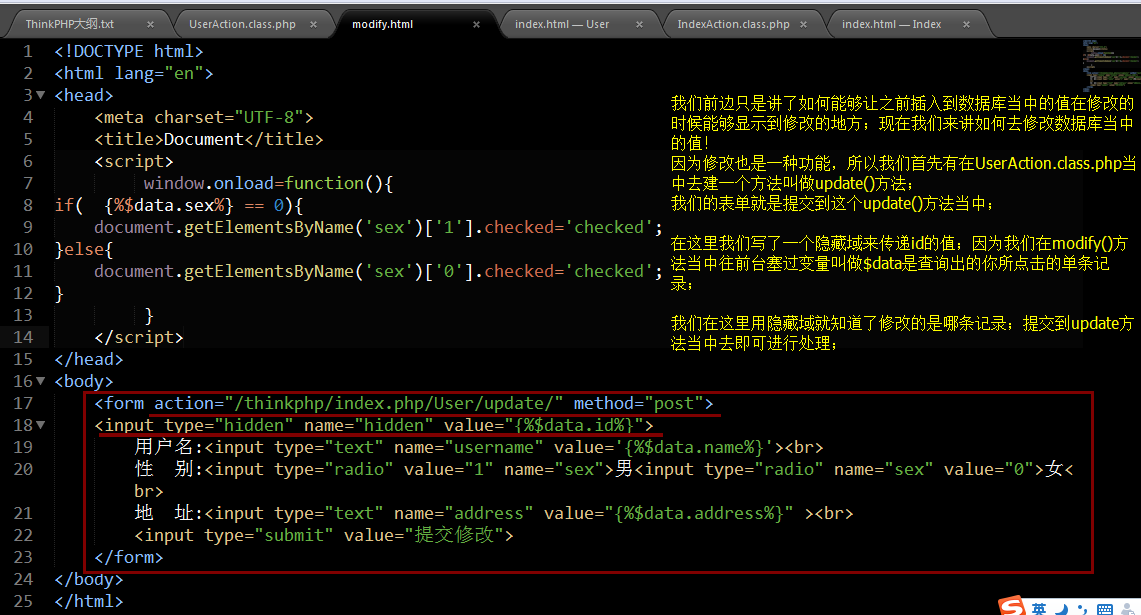
C.最后一把教会了我们如何让单选按钮可以定位到之前我们选择好的那个单选按钮之上：（详情见第四个知识点）



1. ：ThinkPHP.3.1.2.CURD演示2

我们继续上边的来讲修改：（上边我们只是把之前的数据能够让它在修改的地方显示 现在我们开始讲如何修改数据库当中的值）

因为修改也是一种功能 所以我们也要建立一个方法，来访问这个方法即可：

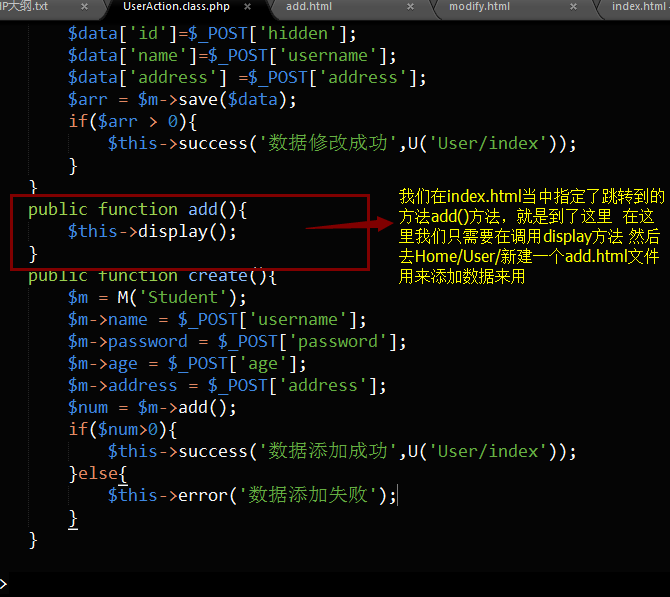


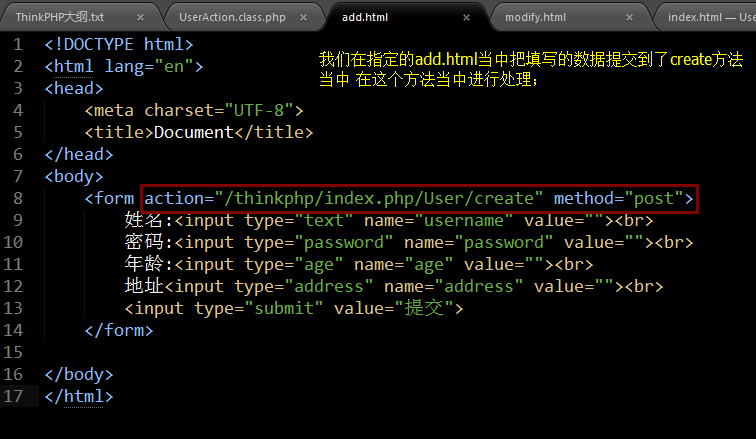


上边讲的是修改数据

那么如何来添加一条新数据呢？





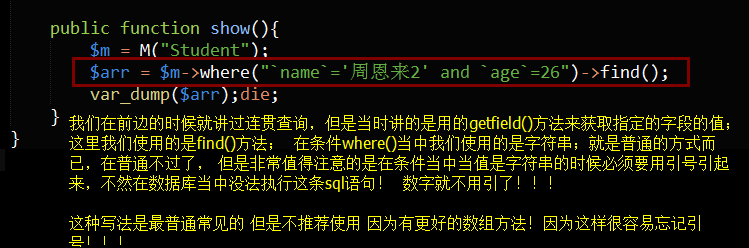




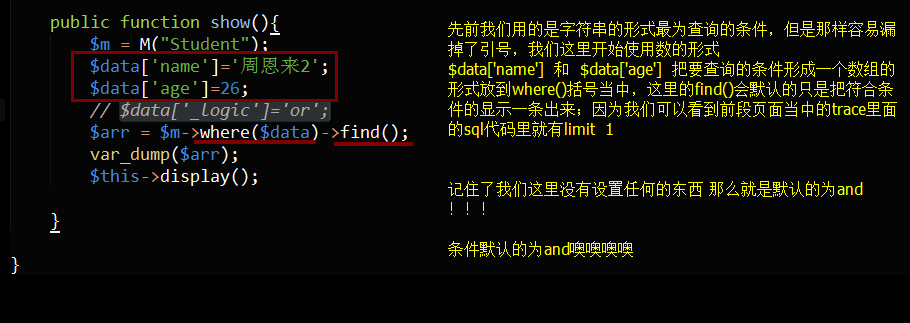
1. 课：ThinkPHP 3.1.2 查询方式的一般使用1
2. 普通查询方法：

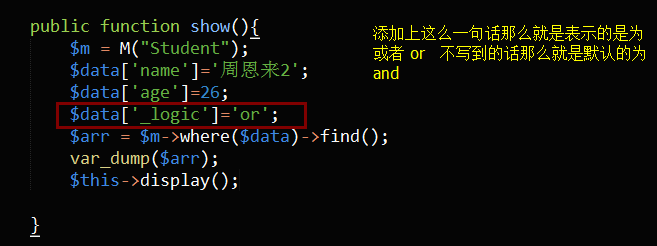
（下边我们使用的都是find 记住了:使用find默认的就是只能获取到一条数据，要是想要获取到所有的符合条件的那么就得要使用select（）方法！）

A:字符串形式的：



B:数组形式的:

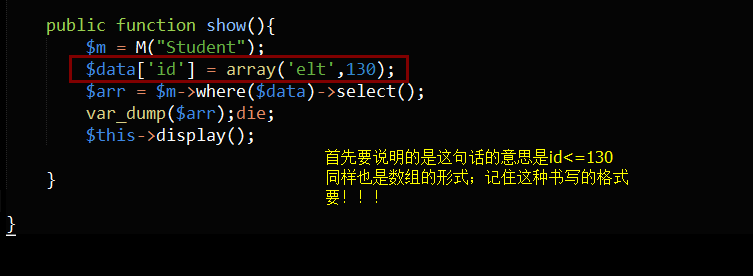




就算是为or 或者了 但是我们这里使用的find（）也只是默认的显示一条记录出来！

1. 表达式查询方式

（当我们要查找的id>30 的时候我们就要用到表达式查询 但是不是区间查询，并不是大于30小于20的这种区间查询方式）



eq 等于

gt 大于

lt 小于

egt 大于等于

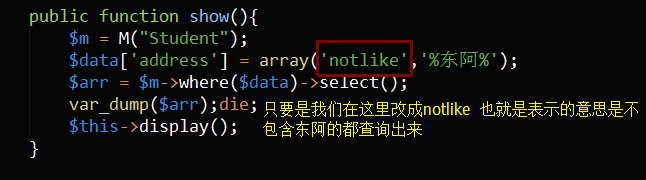
elt小于等于

neq不等于

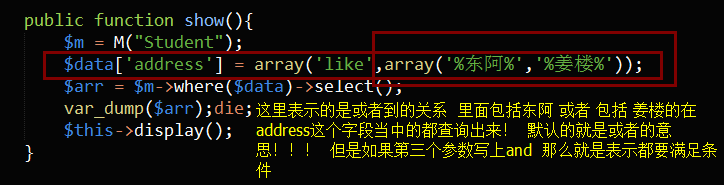
这里讲的只是单一的小于多少 或者说大于多少 ，但是并不是在某个区间当中的！

模糊查询(也是属于表达式查询的一种)（模糊查询某个字段当中包含一些数据的情况下）

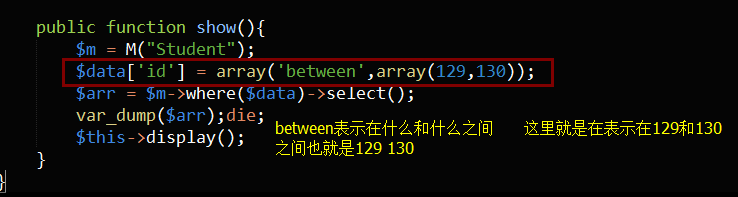
 我们也可以查询出某些字段当中不包含某些数据的情况（使用notlike 这个单词不要中间写空格）



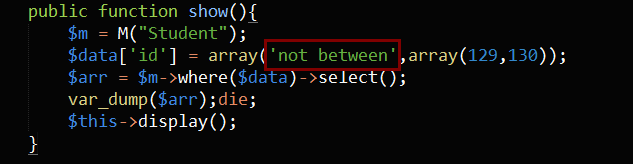
除了上边的两种形式之外还有一种形式就是满足多个的模糊查询：



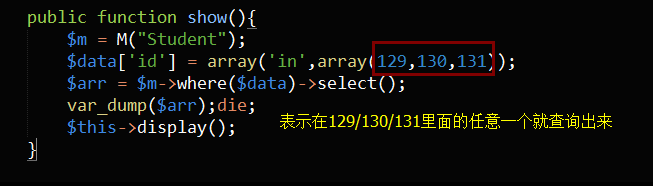
Between的查询方式(在什么和什么之间)



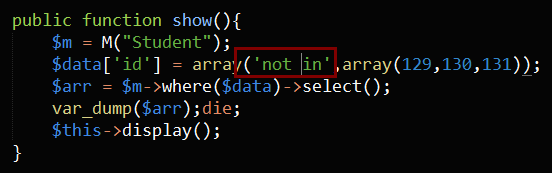
不在什么和什么之间



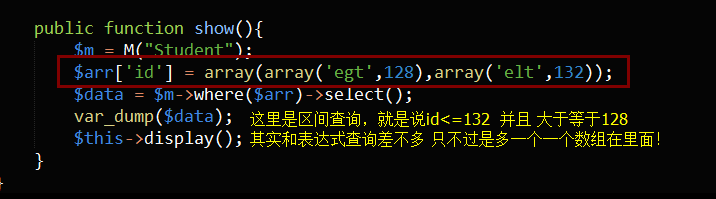
IN的查询（表示在什么里面）



NOT IN 表示不在什么里面的查询出来

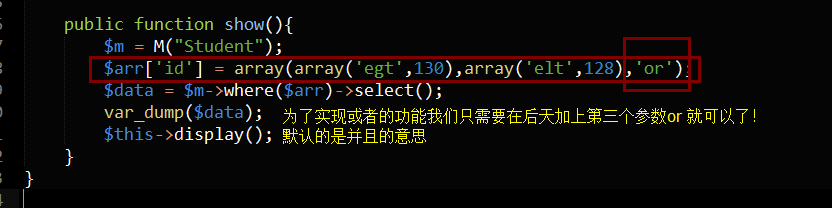


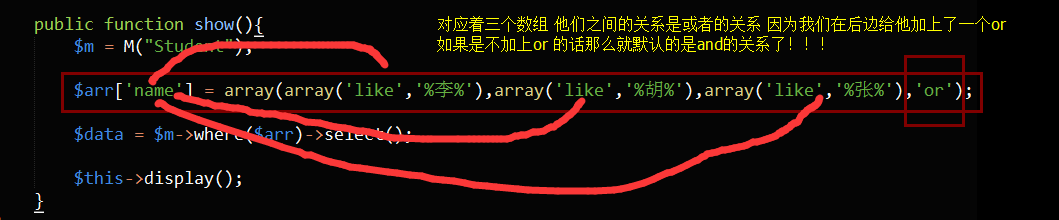
1. 课：9.ThinkPHP 3.1.2 查询方式的一般使用2
2. 区间查询：（大于20并且下雨30的 这样的在某个区间里面的情况）（如果什么都不写那么就是and 并且的意思）



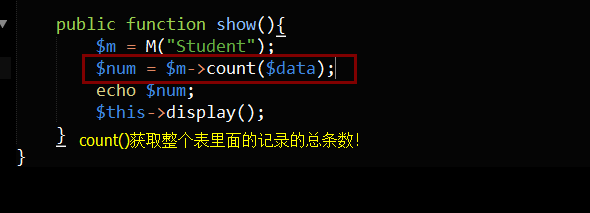
1. 如何才能够实现或者的查询方法呢？

（我们只需要加上第三个参数or就行了）

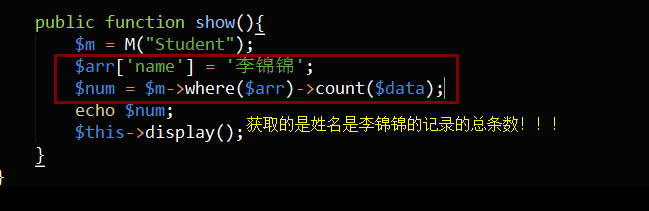




Count:

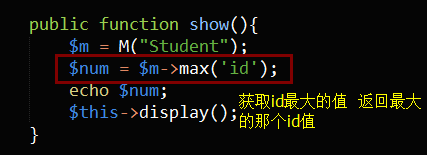


当然我们也可以给count加条件 获取指定的条件下的记录数：

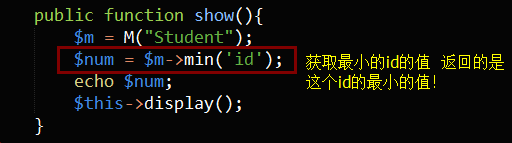


统计查询：

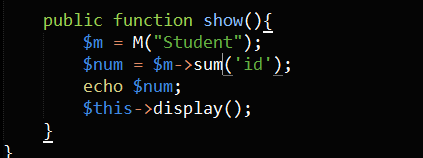
max：（获取某个字段的最大值 比如id）



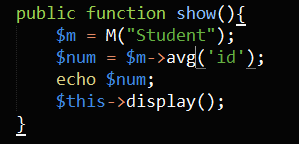
Min:(获取某个字段的最小值 比如id)



Sum（求和）

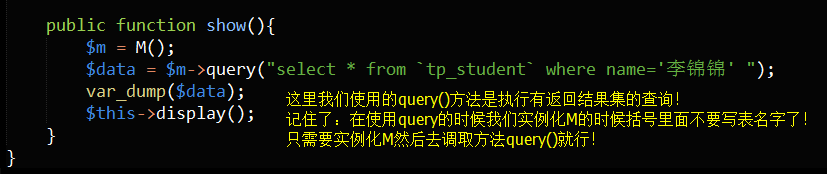


Avg(求平均数)

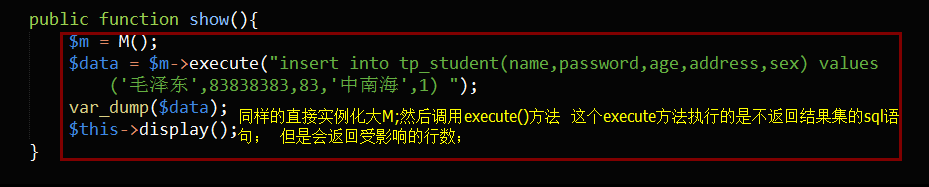


我们上边学习的这些普通方法 表达式方法 区间方式 统计查询 有时候仍然是无法满足我们的查询的需求 这时候ThinkPHP还给我们提供了 sql直接查询：

A: query()主要处理select有返回结果集的查询：（返回查询到的结果）



B: execute()主要是处理insert update这些没有返回结果集的处理：（返回受影响的行数）

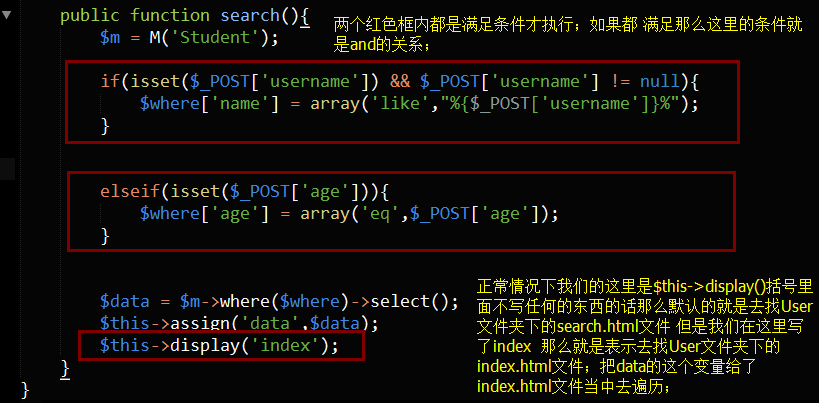


1. :ThinkPHP 3.1.2 查询方式实例演示(主要讲的是搜索功能的实现)

1.



2.



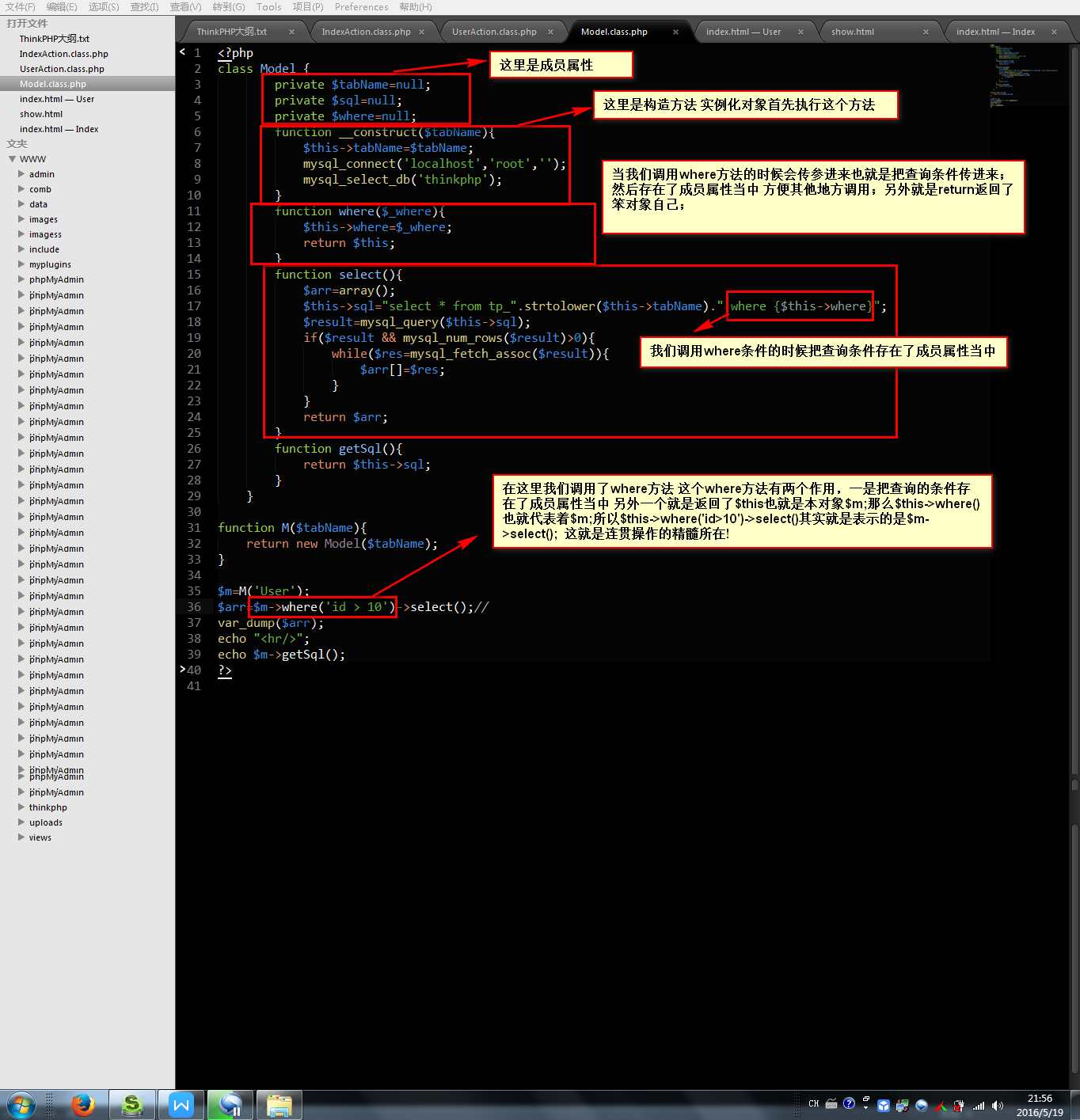
记住一点：

$this->display() 如果括号里面什么都不写那么表示的就是去找所在的控制器当中的这个方法名字相同的html文件;

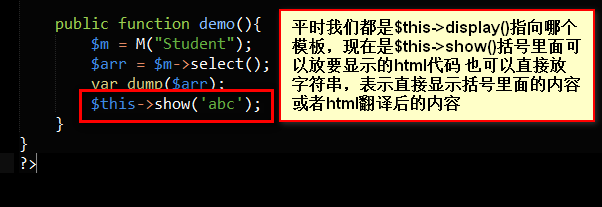
如果写了$this->display(‘index’);那么就是去找所在的控制器假设为User下的index这个方法对应的文件也就是index.html文件；

1. 课:ThinkPHP.3.1.2.连贯操作

1.现在详细阐述连贯操作的原理是什么？



2.$this->show()方法：



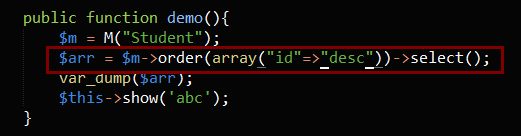
3.常用的连贯操作：

A. where

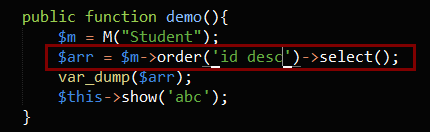
帮助我们设置查询条件(这个前边经常用 不多讲了)

B. Order

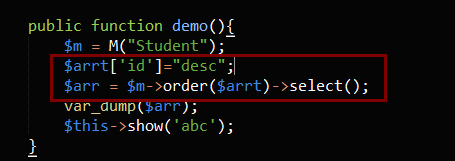
对结果进行排序（三种写法都行）



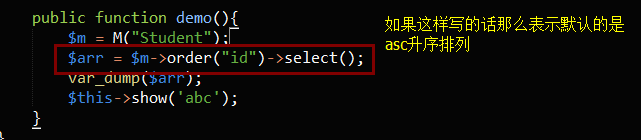
或者



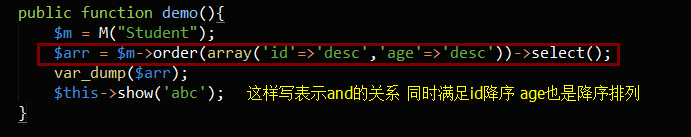
或者



或者



或者



3.limit

限制显示的结果

limit(2,5)

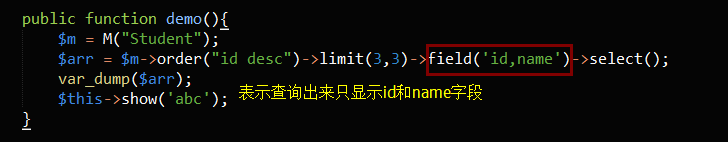
limit('2,5')

limit(10)//limit(0,10)

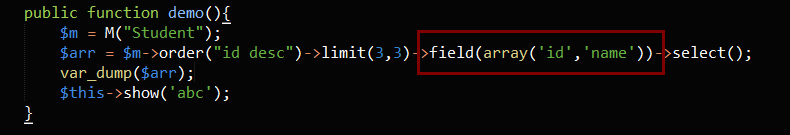


Limit和order 是不分先后顺序的；谁在前谁在后都是一样的 包括接下来将的field(设置查询字段) 都是可以不分先后顺序的！ 但是最后的select（）或者add()等必须在最后边最后边不能动！

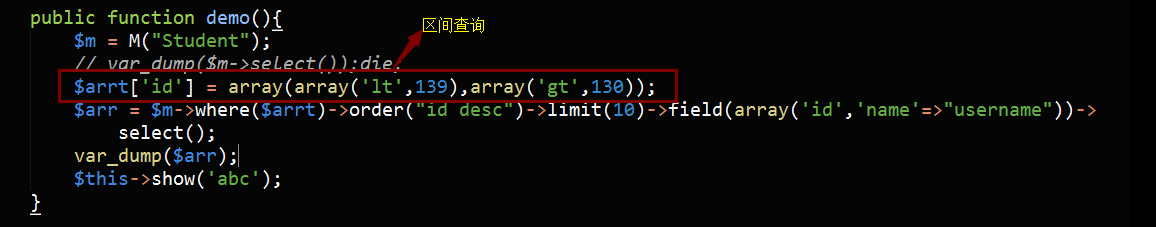
1. field():设置要显示的字段：



也可写成书形式的：



综合例子：



1. 课：ThinkPHP 3.1.2 视图 1

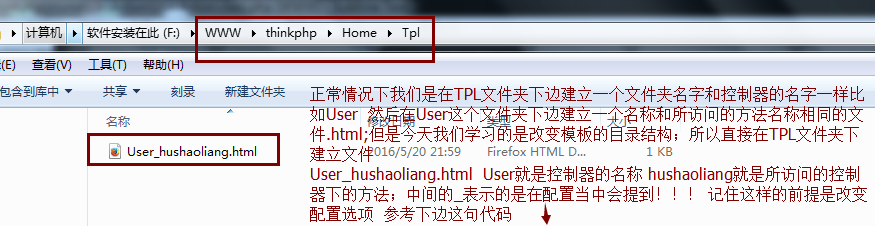
1.规则

模板文件夹下[TPL]/[分组文件夹/][模板主题文件夹/]和模块名同名的文件夹[Index]/和方法名同名的文件[index].html（.tpl）

更换模板文件的后缀名（修改配置文件）

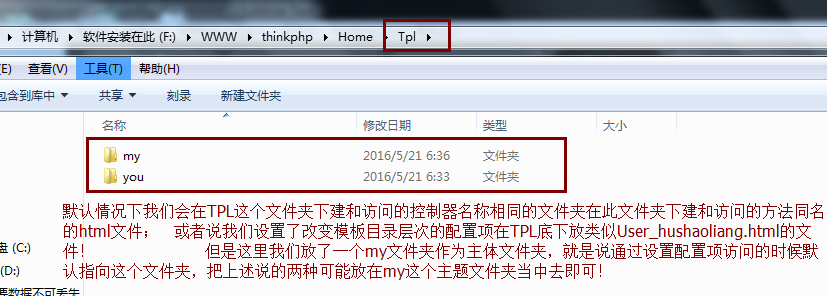
'TMPL\_TEMPLATE\_SUFFIX'=>'.html', //更改模板文件后缀名

1. 修改模板目录层次：



'TMPL\_FILE\_DEPR'=>'\_',//修改模板文件目录层次；

1. 模板主题设置



上边我们建立了模板主题文件夹那么就得要去改变配置项：

'DEFAULT\_THEME'=>'my',//设置默认模板主题 ,默认就会到my这个文件夹下边去寻找要显示的模板！

1. 主题模板还可以更换（有时候我们看到很多的网站上可以更换主题就是下边这个原理）



1. 如何动态修改模板主题？

在后台准备一个功能，修改config.php文件中的默认模板项

通过url传递 t=主题 参数可以修改不同的模板

1. 模板的输出：

a、display

1.display中没有参数

$this->display();

2.可以带参数

$this->display(本模块文件夹下的其他模板文件);

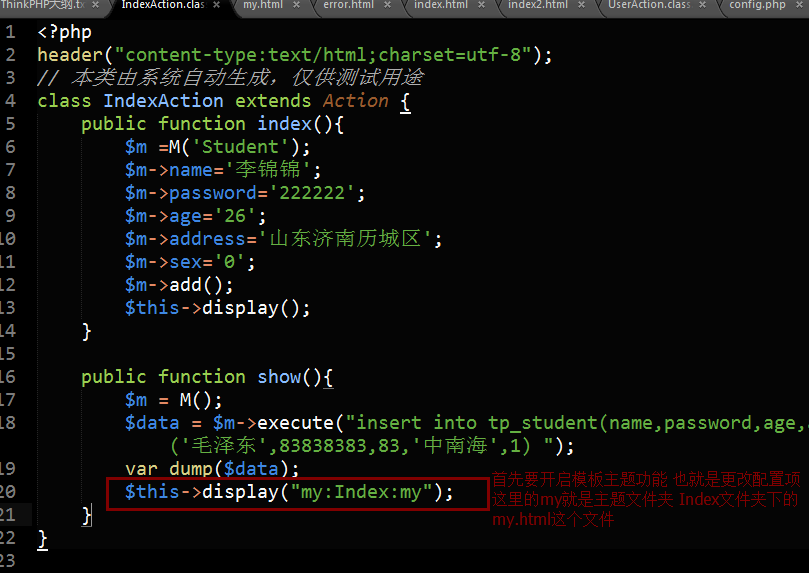
$this->display('index2');（这里的括号里面一定要写index2而不是不是不是index2.html）

3.$this->display(其他文件夹下的模板文件);

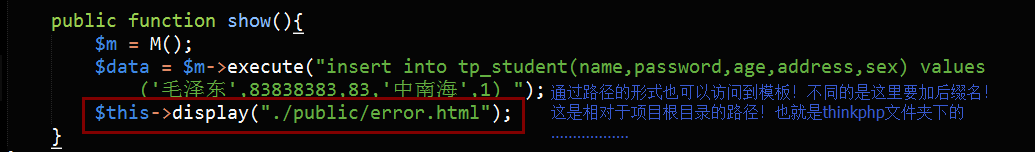
$this->display('Public:error'); //注意，仅仅需要在Tpl 下有 Public文件夹以及其中的error.html即可，不需要一定有Public模块 (这里括号里面直接写Public:error即可)

4.$this->display(其他主题下的 文件夹下的 模板文件);//需要开启主题支持

$this->display('my:Index:index');

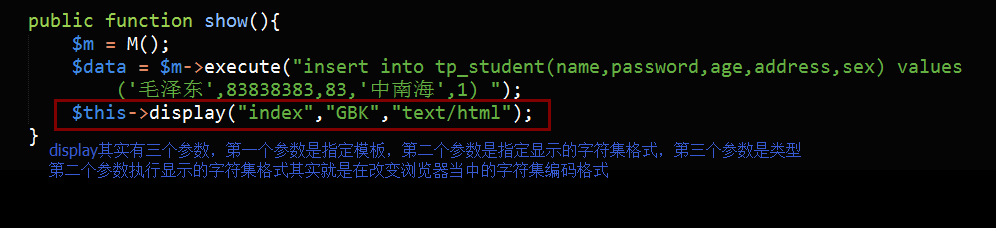


1. 课：.ThinkPHP 3.1.2 视图 2
2. 除了上边第十二节课当中我们讲的通过display来访问模板的方法之外还有一种方法就是通过路径直接来访问模板：

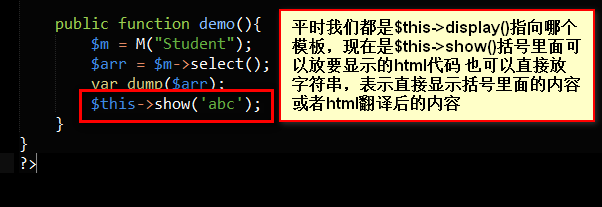


里面的路径是当前文件夹下的，这个当前是当谁的前呢？ 因为我们这里的这个控制器文件都是要在主入口文件当中调用的，所以是当的前台主入口文件的前，也就是index.php文件同级目录下也就是根目录下！

1. 其实display有三个参数 我们前边的学习知识用到了第一个参数来指定模板：



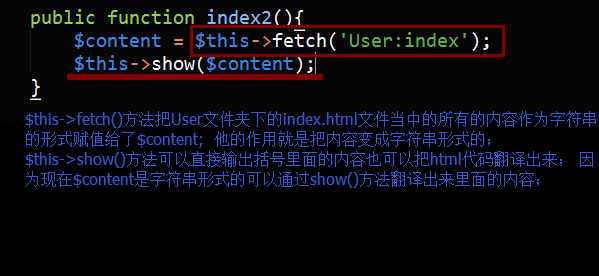
1. 详细介绍$this->show()方法：



1. $this->fetch()方法：

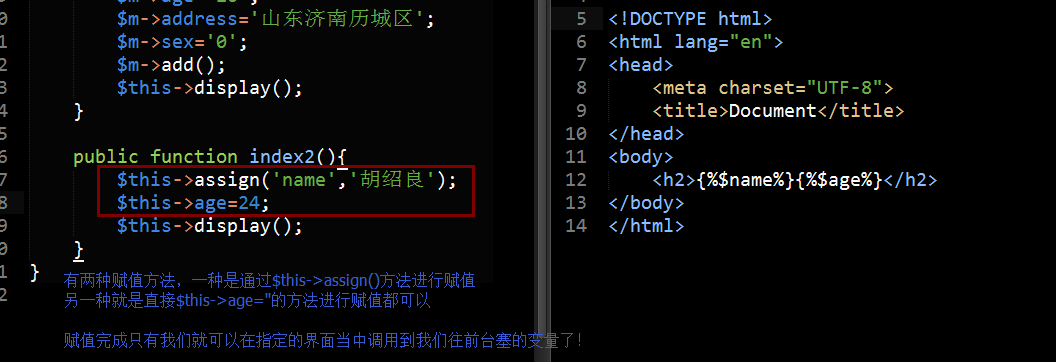
获得模板文件中的内容，以字符串形式返回

$this->show和$this->fetch()两个方法的联合使用案例：



1. 模板中的赋值：

有两种方法：



1. \_\_PUBLIC\_\_：会被替换成当前网站的公共目录 通常是 /Public/

比如：



1. 模板替换：

\_\_ROOT\_\_： 会替换成当前网站的地址（网站根目录）

\_\_APP\_\_： 会替换成当前项目的URL地址（网站根目录/主入口文件）

\_\_GROUP\_\_：会替换成当前分组的URL地址 （不含域名）

\_\_URL\_\_： 会替换成当前模块的URL地址（网站根目录/主入口文件/模块）

(<form action="\_\_URL\_\_/do\_login" method="post">)

比如这里的\_\_URL\_\_/do\_login表示的就是当前模块下的do\_login方法；

\_\_ACTION\_\_：会替换成当前操作的URL地址 （网站根目录/主入口文件/模块/具体哪个方法）?(具体情况下输出看一下 不好解释)

\_\_SELF\_\_： 会替换成当前的页面URL(获取到的是整个的url地址)

1. 我们上边使用的\_\_PUBLIC\_\_表示的是当前网站的根目录，那么我们可不可以自定义一些这样的模板替换变量呢？当然可以

修改配置项文件config.php

更换模板变量规则，修改配置项(通过\_\_CSS\_\_就能直接调取到指定路径下的文件了)

'TMPL\_PARSE\_STRING'=>array( '\_\_CSS\_\_'=>\_\_ROOT\_\_.'/Public/Css',

'\_\_JS\_\_'=>\_\_ROOT\_\_.'/Public/Js',

),

1. 课：ThinkPHP 3.1.2 视图实例演示-登录页面
2. 首先介绍验证码：

验证码的实现需要加载扩展文件，也就是刚开始老师讲的thinkphp扩展包；我们需要把解压后的扩展包里面的Extend文件夹里面的所有内容全部复制，然后找到项目当中的核心的thinkphp文件夹，这里说的是thinkphp核心文件夹哦！然后找到thinkphp核心文件下的Exetend文件夹 里面会是只有一个文件的空文件夹，然后把复制的那些文件全部拷贝到这里面，这也就是thinkphp的扩展库；我们通过手册就可以看的出：

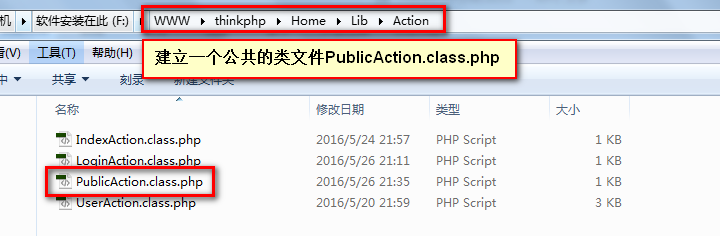
Public function verify(){

import('ORG.Util.Image'); //引入扩展库当中的ORG/Util/Image类

Image::buildImageVerify();//调用Image类下的这个方法

}

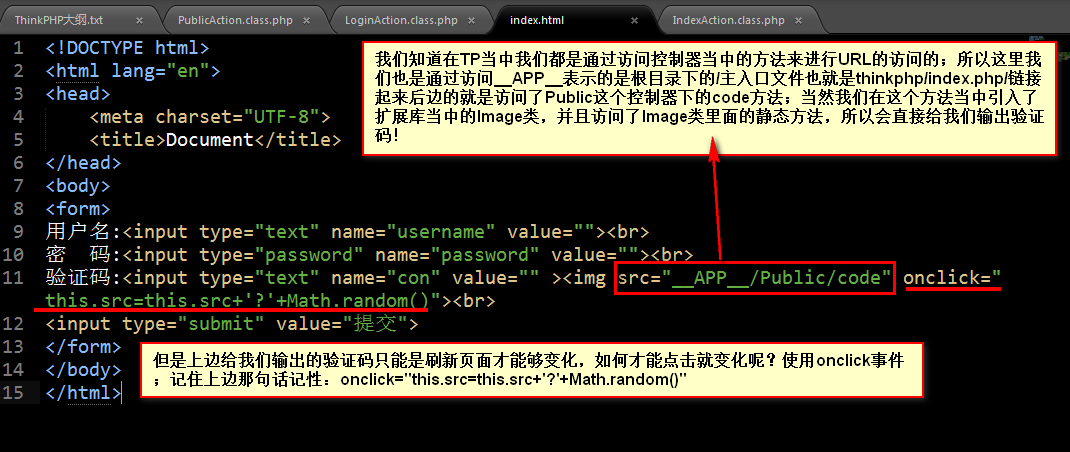
所以我们只需要建一个公共的类文件就行了：



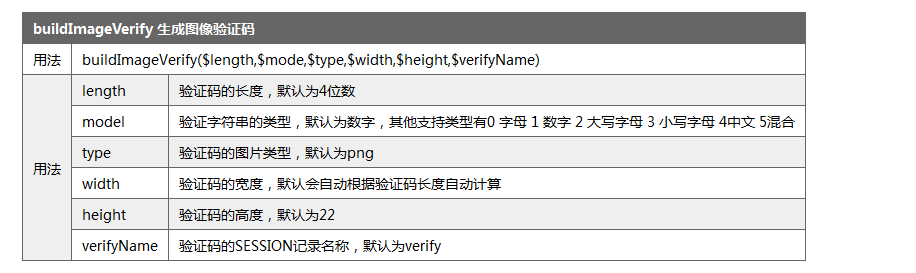
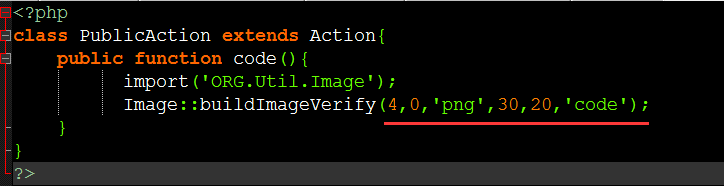
文件当中的内容是：



那么如何调用呢？（我们在调用验证码的同时验证码是被写入到了session里面的）



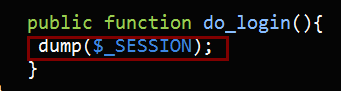
这样就实现了验证码的效果!

Image::buildImageVerify()在括号当中我们可以传进去很多的参数 详情参照手册验证码：  
  


2.深刻剖析\_\_URL\_\_是啥意思？

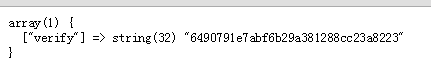


1. 在TP当中我们是使用dump（）来打印输出的！

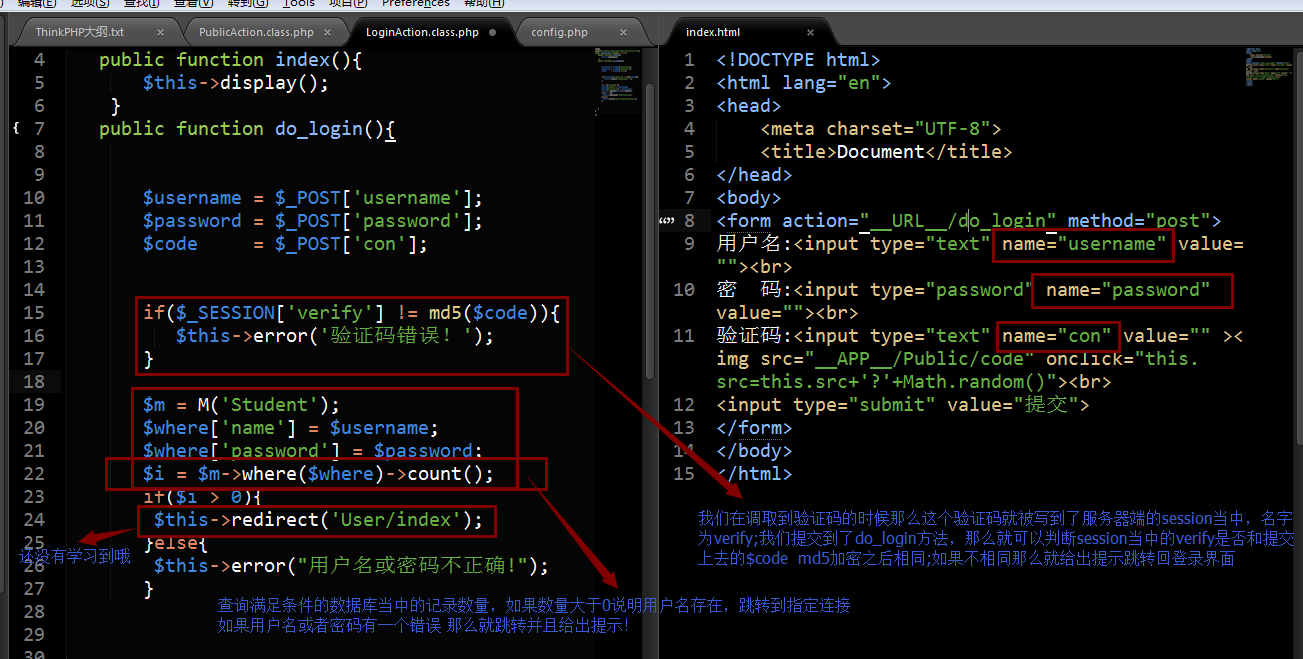


我们可以打印出存储在session当中的验证码的加密了的值

1. 我们在调取验证码的时候验证码就已经被用md5加密写在了服务器当中的session里面了；我们可以dump($\_SESSION)可以看的到服务器当中的session验证码！

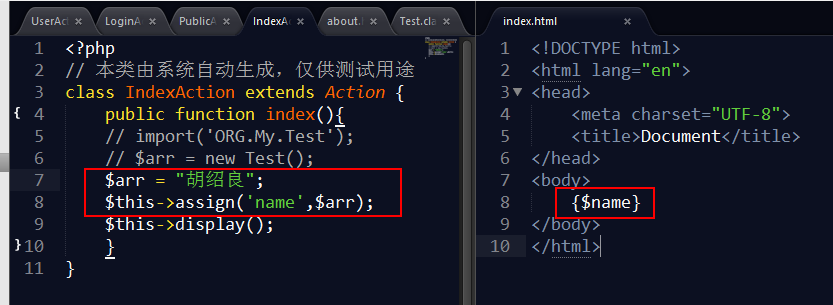
下图就是打印出来的存储在session当中的加密了的验证码：

5.登录小案例：



1. 课：模板变量详解
2. 变量输出：

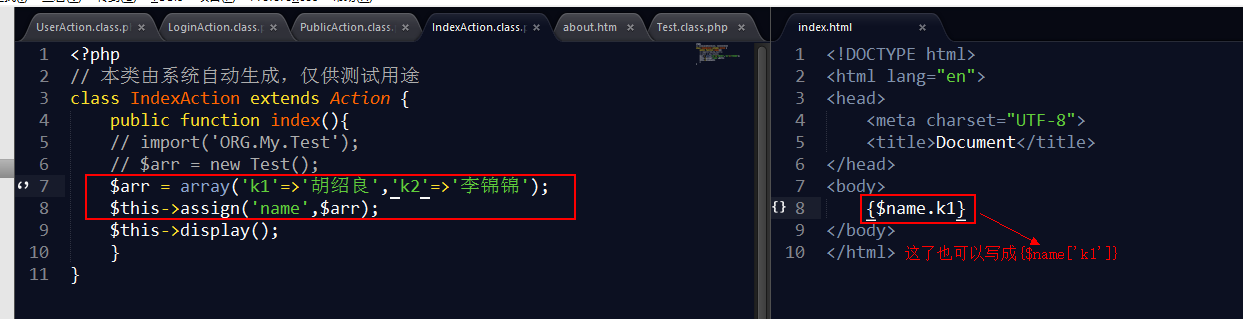
A：标量输出：php当中的四中标量（整形 字符串 浮点型 布尔型）都支持：



B：数组输出：

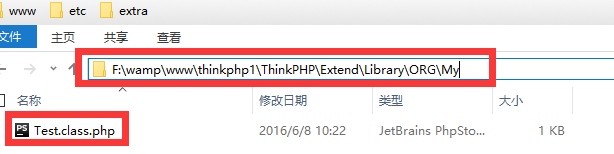
索引数组：{$name[1]}

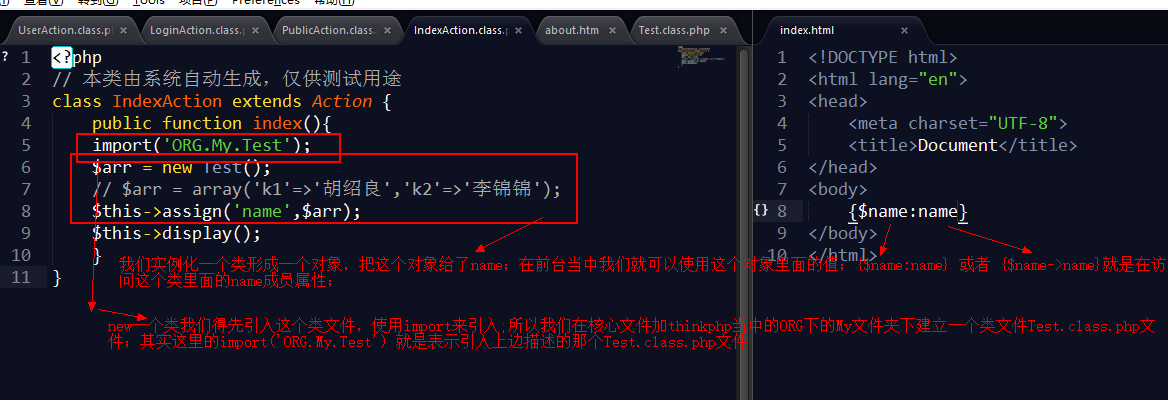
关联数组：{$name['k2']} 或者写成 {$name.k1}



C:对象输出:

thinkphp核心文件夹下的扩展文件夹Extend/Library/ORG/新建一个文件夹存放要引入的类文件比如是My/新建一个类文件即可比如Test.class.php





1. 系统变量：

{$Think.server.script\_name} // 输出$\_SERVER['SCRIPT\_NAME']变量

{$Think.session.user\_id} // 输出$\_SESSION['user\_id']变量

{$Think.get.pageNumber} // 输出$\_GET['pageNumber']变量

{$Think.cookie.name} // 输出$\_COOKIE['name']变量

{$Think.const.HU} //输出常量HU的值

{$Think.now } //现在时间

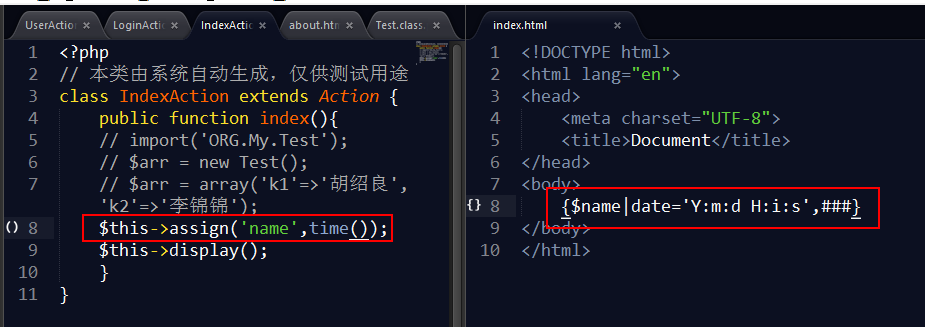
{$Think.post.name}//获取$\_POST当中的name的值

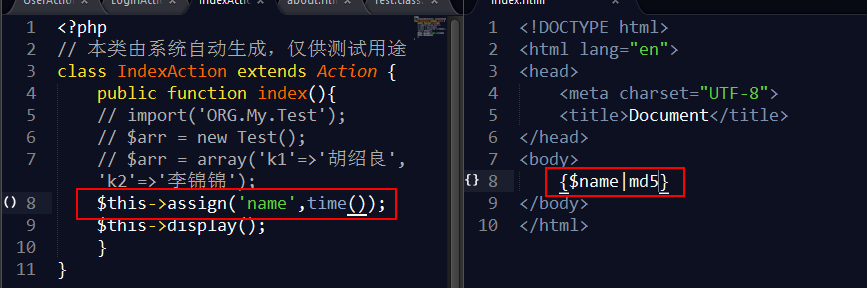


参考手册

1. 使用函数：(参考手册使用函数)

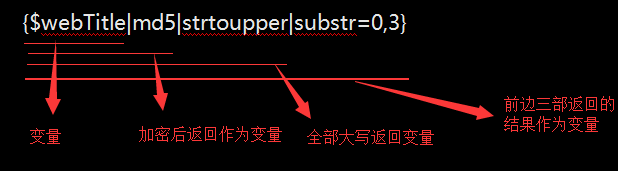
我们可以在视图模板当中使用PHP函数，如果这个函数先来其他的值在给它写变量的话 那么我们使用###来表示占位：比如这里的{$name|date=’y:m:d H:i:s’,###}





小例子：

{$webTitle|md5|strtoupper|substr=0,3}

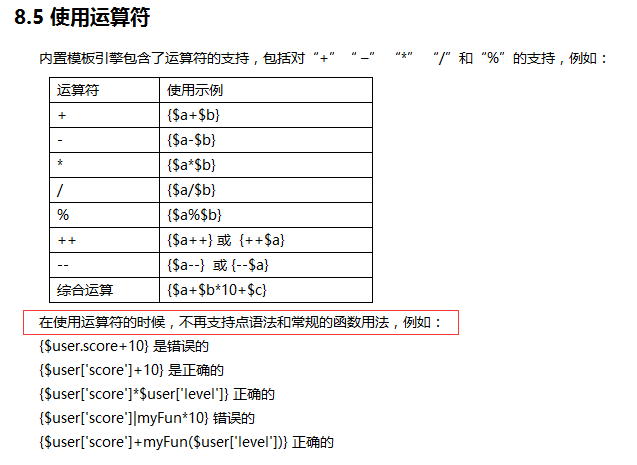


1. 默认值：

{$变量|default="默认值"

(和smarty还是有区别的，smarty当中是default:””但是TP当中是等号)

1. 运算符：



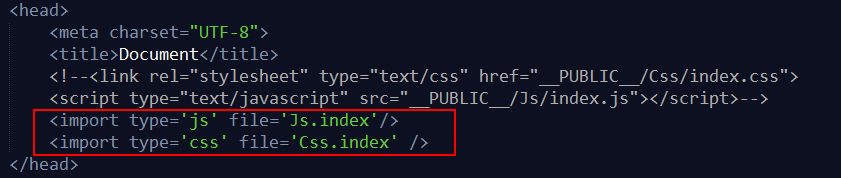
第十六节课：模板当中的语法知识

1. 引入外部css和js文件：

A：（我们前边学习过\_\_PUBLIC\_\_表示的默认就是指向根目录下的Public文件夹 这里的\_\_PUBLIC\_\_/Css/index.css就是指的是根目录下的Public文件夹下的index.css文件）



B：我们使用import引入外部的js和css



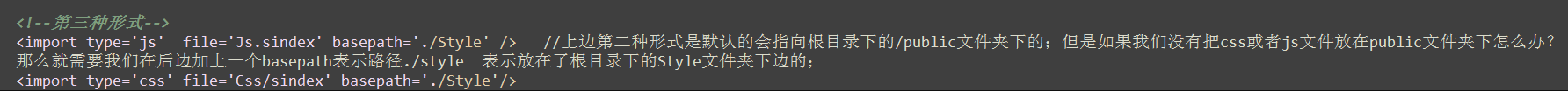
注意：<import type=’’file=’’ /> 这里的import标签默认的类型是js类型；所以当我们在调取外部的js的时候我们可以不写这里的type=’js’ 但是如果是其他的比如css那么就必须要写type=’css’ 了！！！

注意： import标签和\_\_PUBLIC\_\_都是默认的指向根目录下的public文件夹下！我们注意到file=‘Js.index’ 这里的Js表示的是根目录下的public下的Js目录，然后就是个点表示的是/的意思；也就是说Js.index其实就是Js/index.js

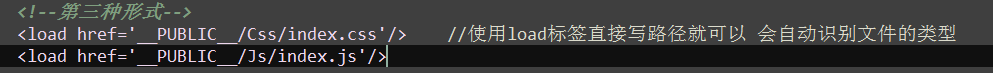
但是还有一点就是如果我们没有把js或者css文件放在根目录下的Public文件夹下改怎么办呢？比如放在了根目录下的Style文件夹下；那么我们就需要给<import >标签里面加上一句代码：

<import type=’js’ file=’Js.index’basepath=’./Style’/>

如图所示：

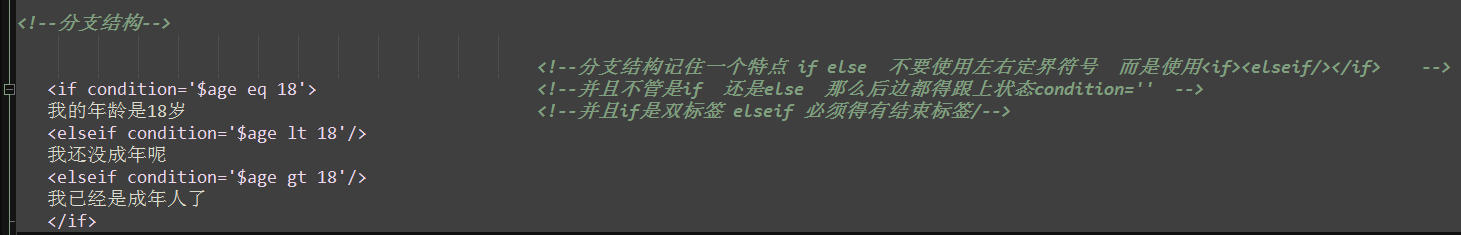


C:使用load标签引入js或者css



直接写文件路径即可

2．分支结构：



分支结构注意的几个地方：

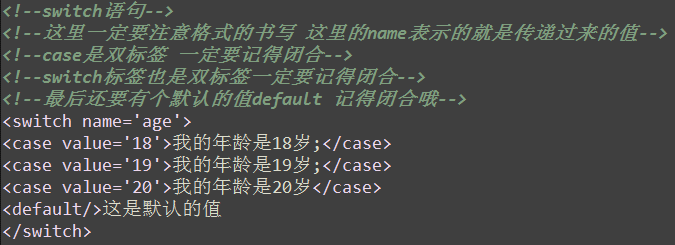
A：if else 或者是if elseif elseif else 等等不同的结构 都不能使用左右定界符 而是使用<>来进行闭合 这一点和在smarty当中是完全不一样的；

B：<if></if>闭合一定要闭合

C：elseif属于单标签 也必须要闭合 <elseif condition=’’/>

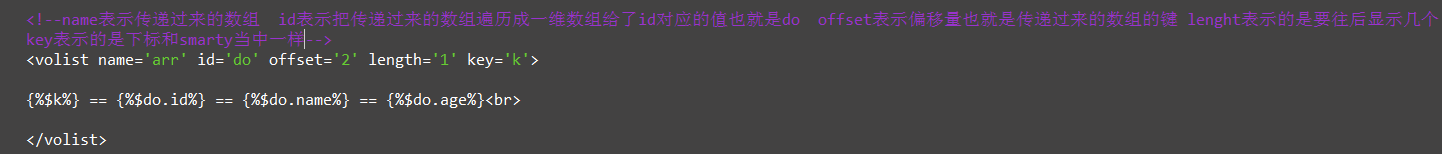
d:不管是if 还是else 后边必须跟上状态condition=’’  
<if condition=’’><elseif condition=’’/></if>

3.switch标签：



1. 课：ThinkPHP 3.1.2 模板中的语法知识 2
2. for循环感觉用到的地方不多 直接略过 用的时候去看手册
3. Volist:  
   一维数组：  
   

二维数组：



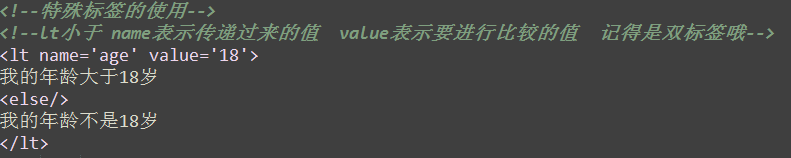
1. foreach: (name表示传递过来的变量 item表示循环出来的一维数组 key表示循环的键值)  
   <foreach name=’’item=’list’key=’k’>

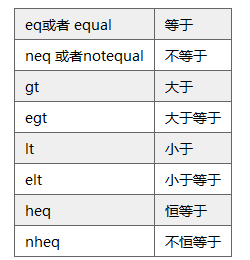
{%$list.name%} == {%$list.id%} == {%$list.age%}

</foreach>

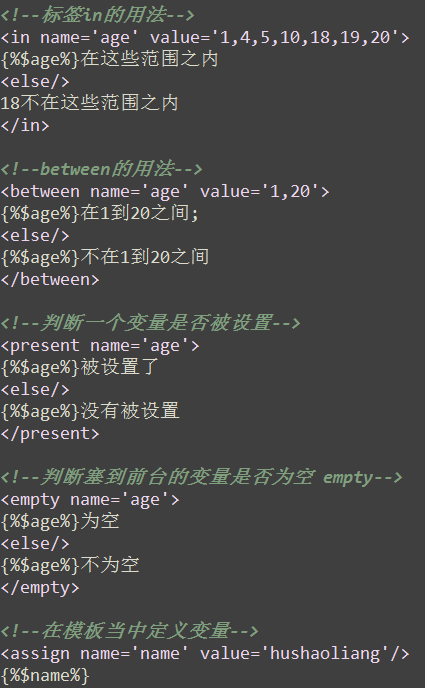
1. 课：ThinkPHP 3.1.2 模板中的语法知识 3
2. 特殊标签的使用：  
   我们在前边见讲过

<if condition=’’><else/></if>

这里讲的特殊标签就是代替<if></if>的标签 我们完全可以用<eq></eq> <lt></lt> <gt></gt>比如下例子：  




这些比较标签都支持上边的写法；

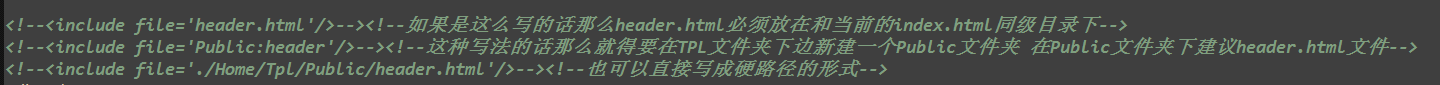


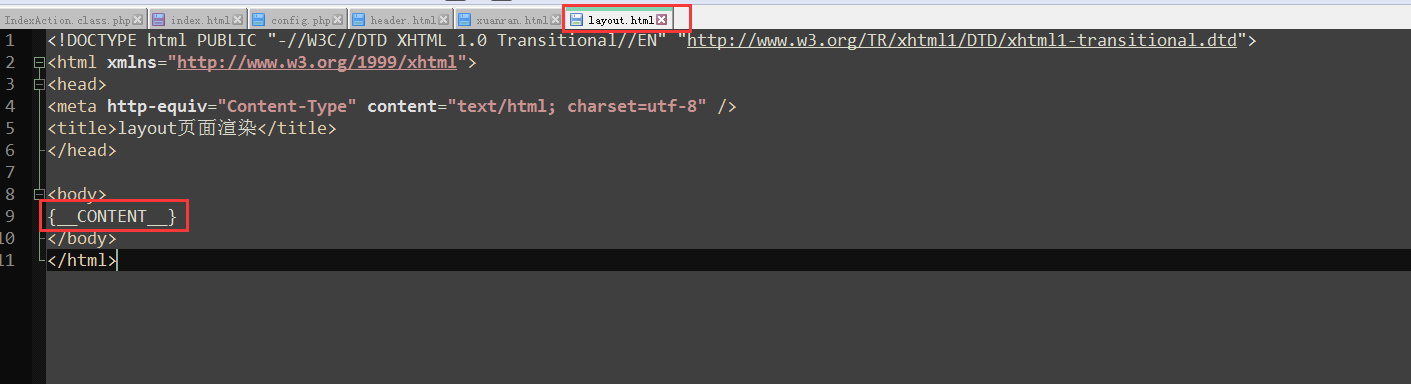
1. 其他标签的使用：

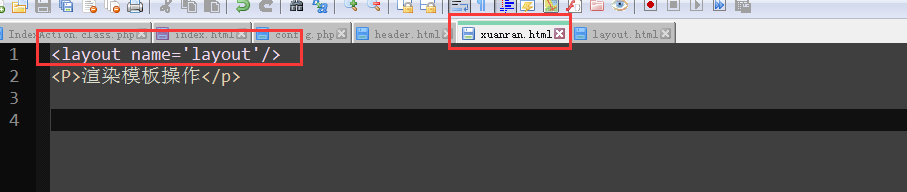
在THinkphp当中的模板里面我们是可以直接写php代码的：（一定要注意格式<php></php>）



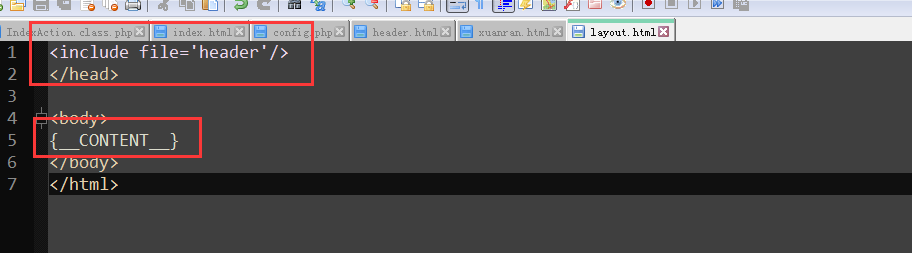
1. 课：ThinkPHP.3.1.2.模板的使用技巧.1
2. 模板的包含：（具体详情去参考手册 常用的就这三种）

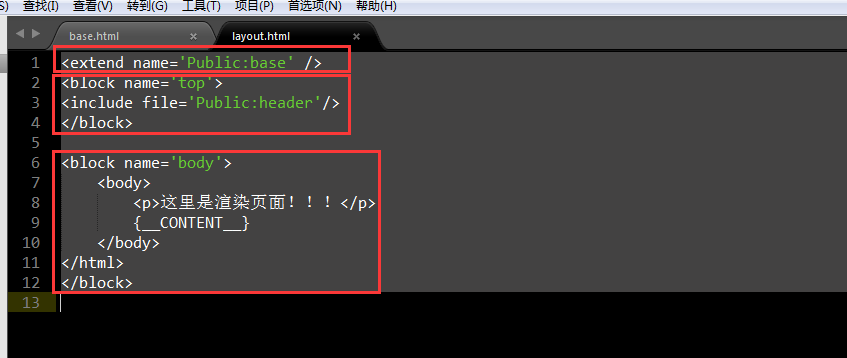


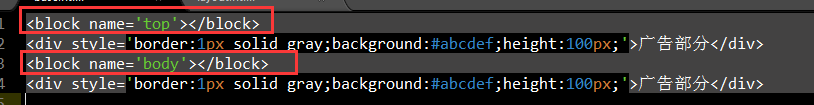
1. 模板渲染：  
   A：我们在Home文件夹下的Tpl模板文件夹当中新建一个文件叫做layout.html;  
     
   使用{\_\_CONTENT\_\_}代替要被显示的内容；

B:在模板当中使用一句话来调取到渲染公共页面：  
  
 渲染模板用最简单的话来描述就是：比如有一个xuanran1.html 有一个xuanran2.html 但是这两个模板的不同之处很小 这时候我们就可以采用模板渲染 就是把两者的公共部分提取出来放到layout.html当中去，然后把这个layout.html放到Home/Tpl/文件夹下；在这个layout.html当中把xuanran1.html和xuanran2.html当中不同的显示部分用{\_\_CONTENT\_\_}来表示；然后在xuanran1.html当中和xuanran2.html当中分别使用<layout name=’layout’/>来调取渲染公共模板就行；

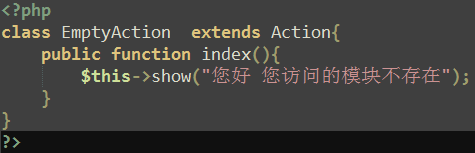
渲染的公共模板必须放在TPL文件夹下才行 调取的时候直接写  
<layout name=’layout’/> 即可；这里的name就表示的是Tpl文件夹下的layout./html文件；

第二十节课：ThinkPHP 3.1.2 模板的使用技巧 2  
1.在渲染模板当中我们也可以使用include包含模板 然后这个渲染模板又被调取到别的模板当中使用的是<layout name=‘layout’/>

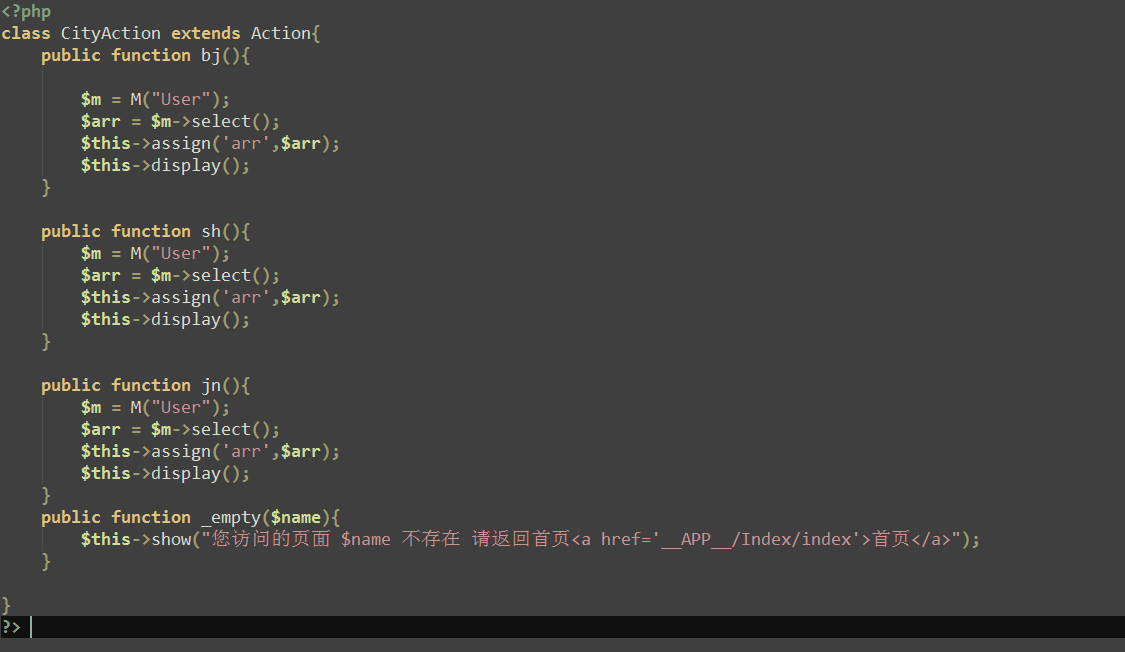
1. 模板的继承：  
   一旦使用了模板继承那么整个的页面的所有的代码都要在不同的<block name=’’></block>标签当中包含起来 如果有的代码没有在这个双标签当中那么在页面当中是显示不出来的 查看源文件也是看不到没有被包含在这个双标签当中的代码；
2. 什么是模板继承：  
   比如这里的layout.html我们要使用很多的<block name=’’></block>区分出很多的块状，然后使用<extend name=’要继承的父类模板的路径’/>  
   

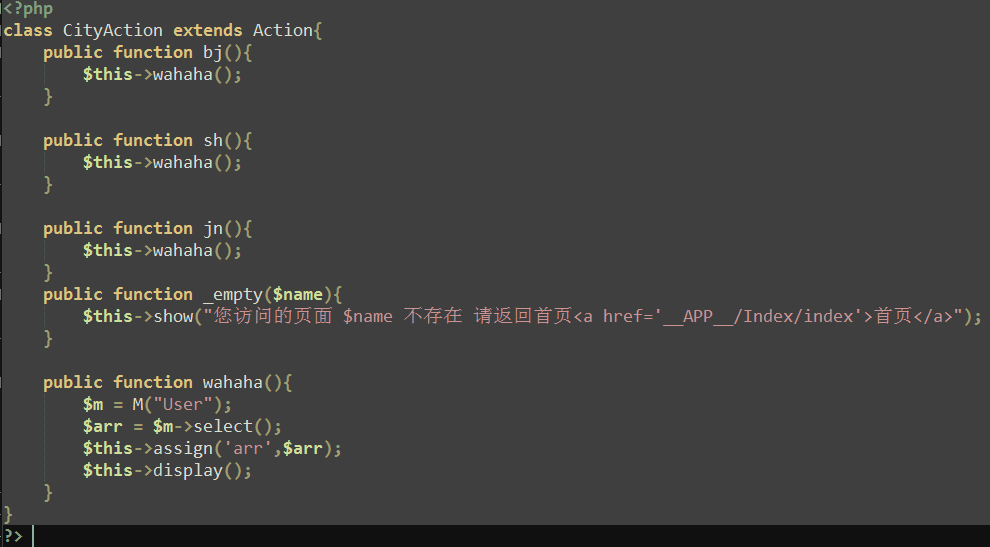
然后我们在父类模板当中这么写：  
  
每个block块当中都会有一个名字 那么在父类模板当中我们只需要写上相同名字的block就表示的是代替了子类模板当中这个块状当中的内容；在代替子类模板当中内容的同时我们可以在父类当中加入我们自己想要添加的内容进去 因为子类当中继承了父类模板<extend name=’’/> 所以我们添加的新的东西会在子类当中显示出来 这就是模板继承的意义所在；

第二十一节课：ThinkPHP 3.1.2 控制器的模块和操作 1  
1.空操作:  
  
本身路径当中City模块下是不存在tj这个方法的 所以也就访问不到那么系统会自动的给我们抛出错误提示：  
  
但是我感觉这样很难看 所以我们可以在访问的模块当中新建一个方法：  
  
取名叫做 \_empty($name)方法 那么当我们访问的这个模块下的某个方法不存在的时候那么就会自动的去寻找这个\_empty($name)这个方法 在这个方法当中就会自动的为我们抛出自定义的错误信息  
$this->show()直接输出到页面当中即可；我们所访问的不存在的那个方法的名字会自动的被括号当中的$name所获取到

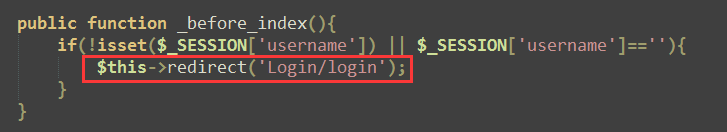
1. 空模块：  
   当我们访问的模块不存在的时候那么系统也会自动的给我们抛出错误提示；那么当我们访问的错误的模块的时候我们让系统自动的去访问一个名称为EmptyAction.class.php的模块； 在这个模块下新建一个index方法；在这个index方法当中抛出错误提示即可；  
   
2. 当访问的模块不存在的时候那么就会默认的去访问这里的这个Empty模块；
3. 系统常量MODULE\_NAME表示自动获取当前的模块名称；在方法当中能够使用到的一些系统常量详情参看手册附录----常量参考

第二十二节课：ThinkPHP 3.1.2 控制器的模块和操作 2

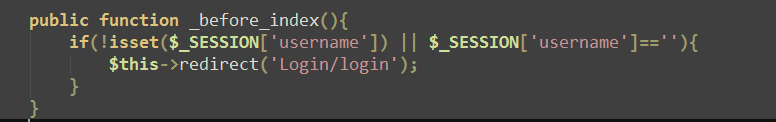
1. 当我们在一个类文件当中的很多的方法当中存在很多的相同的代码的时候 比如下图:  
   

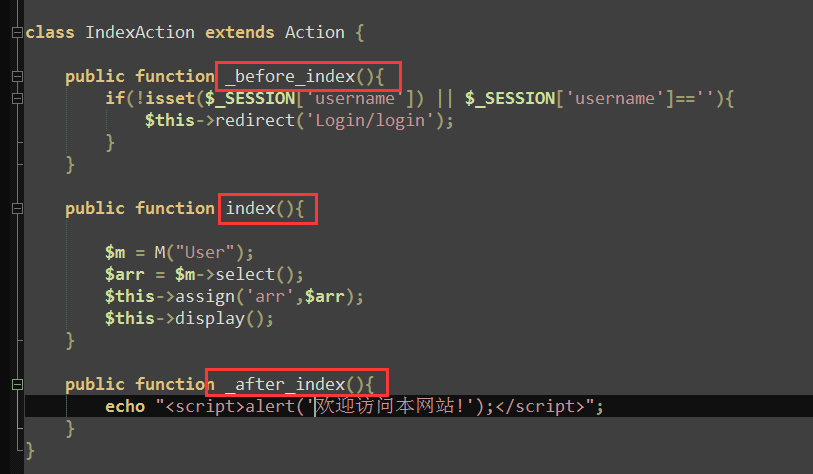
每个方法当中都是一样的代码 那么这个时候我们就可以新建一个方法：  


第二种方法：要不就是再新建一个类文件 然后实例化这个类为对象 然后调取这个类当中的某个方法；

1. $this->redirect(‘User/index’) :  
   直接跳转到指定的页面 没有任何的提示；  
   
2. 前置操作：

前置操作简单的来说就是在我们访问一个模块下的某个方法之前要先访问的一个方法；这个方法的命名就是在我们访问的方法的名字前边加上 \_before\_ 比如我们访问的方法名称是hu()那么我们给前置方法的命名就是： \_before\_hu() 那么这个样子当我们访问方法hu()的时候系统会先自动的去访问 \_before\_hu()方法； 这就是前置操作；  
比如在一个用户访问一个界面的时候我们可以用前置操作对其进行判断是否存在session[‘username’] 从而判断要不要跳转或者其他操作：



1. 后置操作：  
   后置操作就是执行完我们访问的方法之后 会自动的去访问\_after\_名称()的方法；  
   

首先回去访问前置操作 然后访问我们所访问的index操作 最后会自动访问后置操作；

第二十三节课：ThinkPHP 3.1.2 URL 1  
1.URL默认是区分大小写的；

1. 在配置文件当中config.php当中修改配置文件：  
   'URL\_CASE\_INSENSITIVE'=>true, 写上这一句话那么就表示URL不再区分大小写；
2. URL规则：

当我们访问的模块名称太长为两个单词组合的并且首字母都是大写的时候比如:

UserGroupAction.class.php   
那么  
在区分大小的前提下是：  
localhost/index.php/UserGroup/index  
但是在不区分大小的前提下是:

localhost/index.php/UserGroup/index  
或者  
localhost/index.php/user\_group/index (记得在两个单词之间加上一个\_即可)

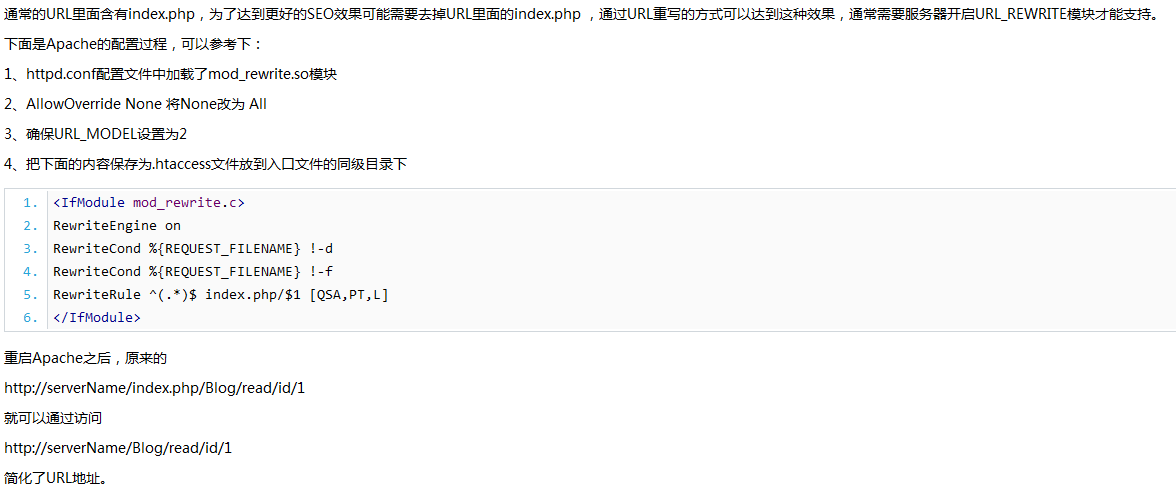
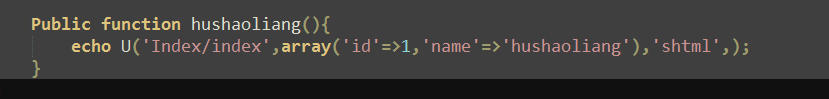
注意：上边只是针对名称为连个单词组合的并且首字母都是大写的情况

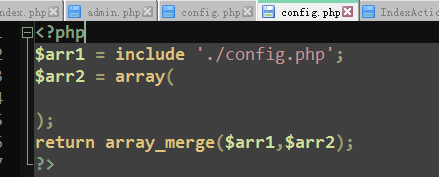
1. URL伪静态：  
   我们在访问模块当中的方法的时候默认都是没有后缀的；TP默认设置了URL伪静态 我们可以通过config.php配置文件进行修改:  
   'URL\_HTML\_SUFFIX'=>'html|shtml|xml',//限制伪静态的后缀 里面的任何一个都可以；

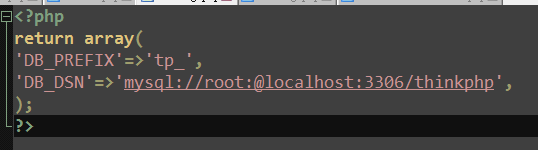
当我们没有在config文件当中设置伪静态的时候我们访问模块当中的方法是这样的：  
但是当我们在config文件当中设置了伪静态之后那么是这样的:

 自动的在尾部加上了.html  
  
5.URL路由设置不明白怎么搞？？？

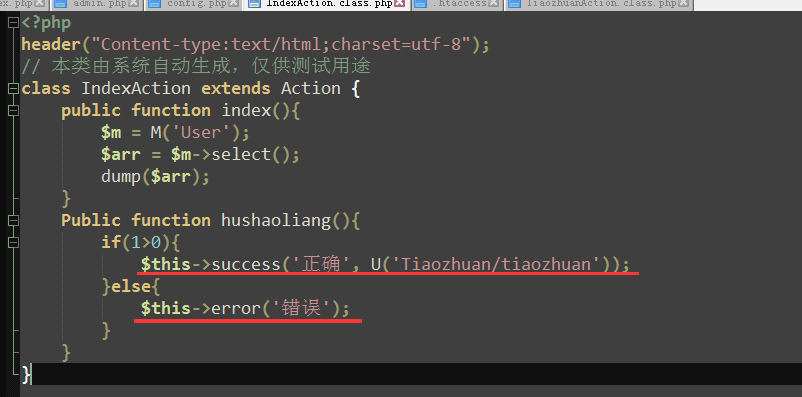
第二十四节课：ThinkPHP 3.1.2 URL 2

1. URL重写：所为的URL重写就是把URL当中的index.php去掉但是还能正常访问：（具体步骤按照手册当中的来操作就行）  
   会生成一个.htaccess的文件放在了和index.php同级目录下  
   
2. URL生成：  
   U方法：  
     
   第一个是模块/方法名 第二个是一个数组 表示的是参数 第三个是伪静态后缀  
   （具体情况参考手册URL生成）

第二十五节课：  
1.多应用配置技巧：  
不仅前台应用当中要配置config配置文件 后天的应用Admin当中也要配置config配置文件 但是对于两个都有的一些配置选项都配置的话会很啰嗦 所以我们在根目录下新建了一个config.php文件专门放前后台配置文件当中共有的部分；在前后或者后台的配置文件当中包含这个共有的根目录下的config.php配置文件 然后再写一个数组专门放前后台自己独有的一些配置内容 最后使用array\_merge()把两个数组合成一个数组返回：  


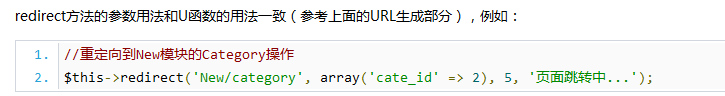
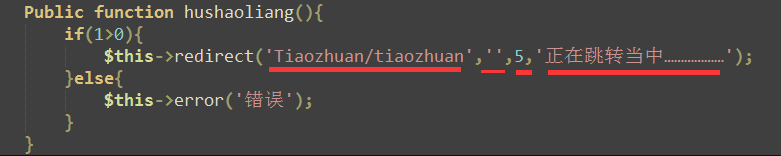
公共的配置文件：  
  
2.使用分组：(不建议使用 根据个人爱好 参考手册)

1. 页面跳转：

  
$this->success(‘提示语句’，‘跳转地址’); 返回成功的这句代码必须给写第二个参数跳转地址；如果是希望跳转到本模块下的其他的方法那么直接写方法的名字就可以了 默认也是跳转到本模块下的某个方法 但是希望跳转到其他模块下的某个方法 那么就得要使用U方法了；因为在U方法当中我们只要输入模块名称/方法名那么就会自动帮我们转换成实际的链接地址；

$this->error(‘直接写提示语句就行’); 自动跳转到上一页面

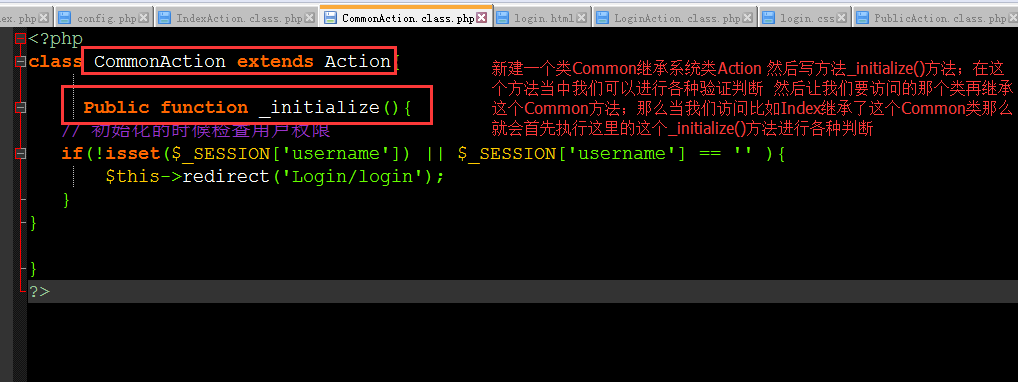
默认的成功提示页面和失败提示页面是在TP核心文件夹Thinkphp/Tpl/dispatch\_jump.tpl文件当中显示；但是我们也可以修改成我们项目内部的文件来显示错误提示：  

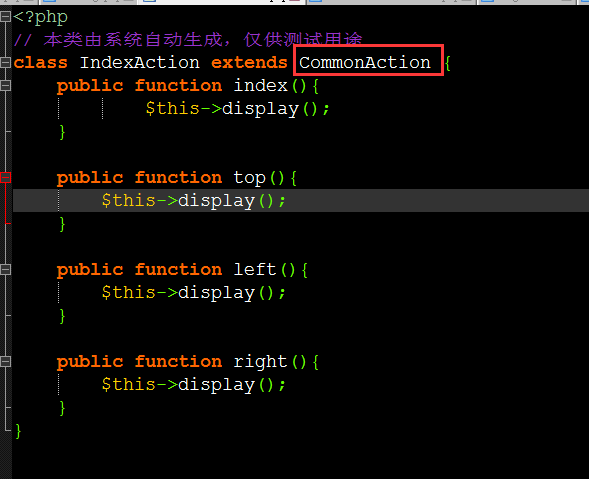

重定向：  
  
  
Ajax返回未掌握？？？？？

项目实战总结：  
1.标签库驱动类：作用就是把某个标签定义解析成为有效的模板文件；比如我们调取验证码<input type='text' name='code'/><img src='\_\_APP\_\_/Public/code/w/30/h/30' onclick='this.src=this.src+"?"+Math.random()'/> 这么长的一段代码我们可以通过标签驱动类文件把它转变成一个比这更简单的一句话来代替它 当我们每次用到它的时候那么我们就直接调取标签库驱动类当中的这个方法就行：<Message:code width='100' height='50'/>  
A:在F:\wamp\www\message\ThinkPHP\Extend\Driver\TagLib目录下新建一个驱动类文件TagLibMessage.class.php 前边必须是TagLib驱动类名称.class.php  
B:  
  
前天调取样例：  

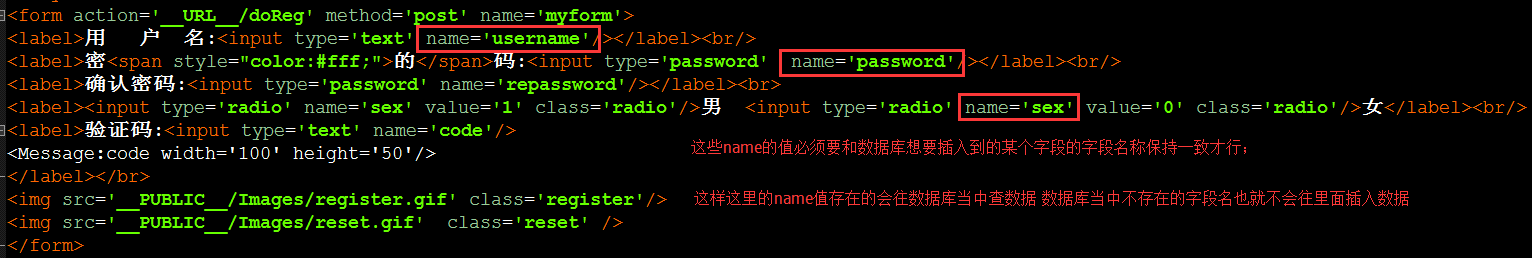
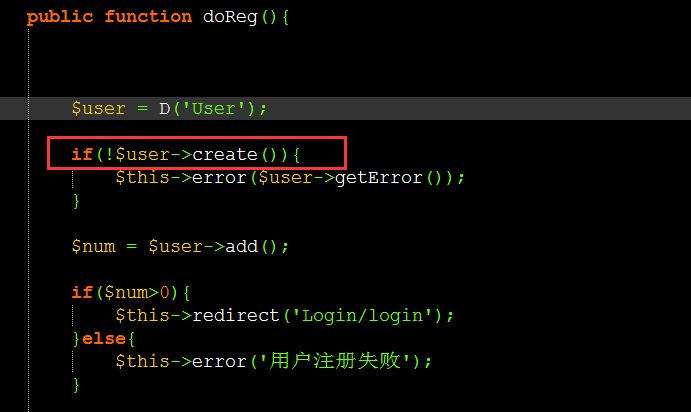

1. 控制器扩展接口：（可以进行权限判断）

我们平时所写的类都是继承与Action系统类 这个Action系统类为我们提供了一个初始化方法\_initialize接口；\_initialize方法会在所有操作方法调用之前首先执行；（比如我们要判断session当中是否存在username 如果存在那么就可以登录不存在就不能登录！）  
那么我们可以写一个CommonAction类让这个类继承Action类；然后在这个CommonAction类当中写一个方法叫做 \_initialize 方法；在这个方法当中进行判断是否存在$\_SESSION[‘username’]  
然后我们可以让INdexAction.class.php当中的Index类继承这个Common类；那么也就是继承了Action类；在当我们使用Index类当中的任何的方法的时候都会自动去首先执行\_initialize()方法；



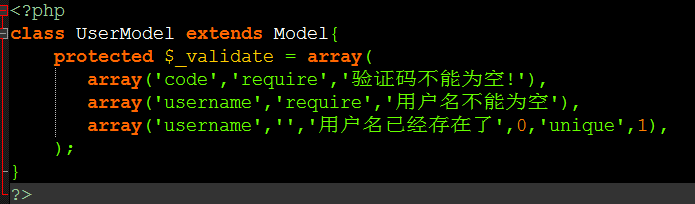


TP当中有三大自动：  
自动创建：（往数据库当中插入数据）  
平时我们往数据库里面插入数据都是从后天$\_POST[]一些值赋值给变量 然后再往数据库当中插入数据 但是这样太麻烦了；TP为我们提供了自动创建数据的方法：

首先  
第一点：我们必须是通过表单提交数据 并且方式是post传参；  
第二点：表单当中的name的值如果想要插入到数据当中去那么必须和数据库当中保持一致才行：  
  
   
自动创建数据会有返回值；

自动创建之后会有返回值保存在了对象$user里面 所以我们可以打印出dump($user)来看一下数据也可以的；

自动验证：（后台当中自动验证提交过来的数据）

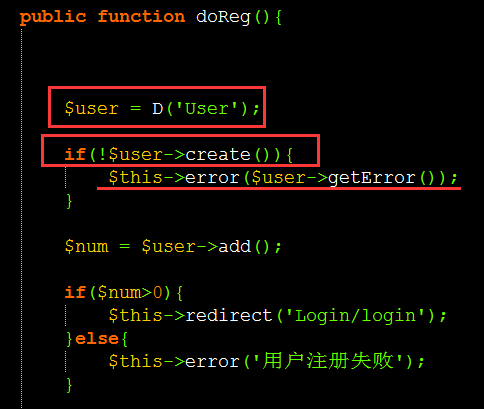
自动验证我们需要写一些规则，但是我们又不能修改系统自带的Model类 所以我们只能在 Home/Lib/Model/文件夹下新建一个model类命名为UserModel.class.php 并且必须以这样的形式来命名；  
首先新建的这个类必须先继承Model类；然后在这个类当中进行规则的编写：  


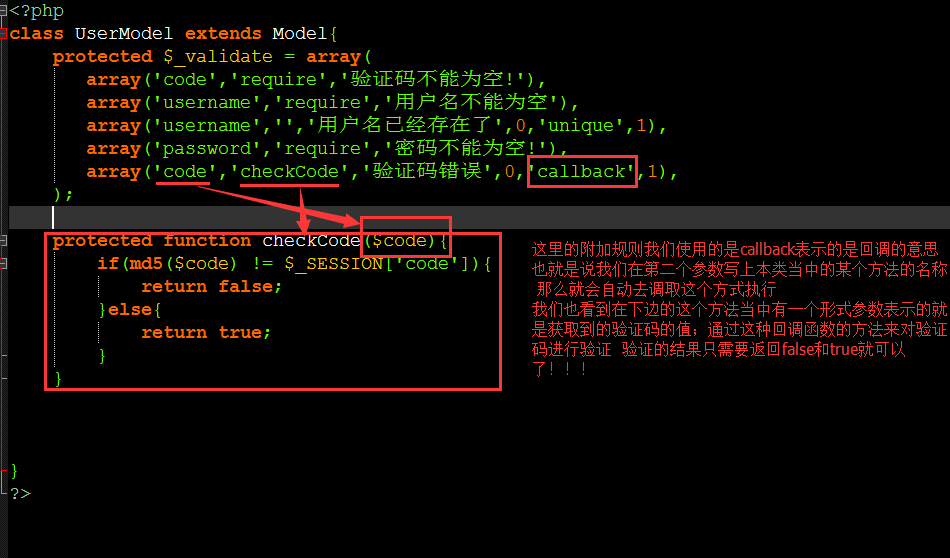
样例：

array(验证字段,验证规则,错误提示,[验证条件,附加规则,验证时间])

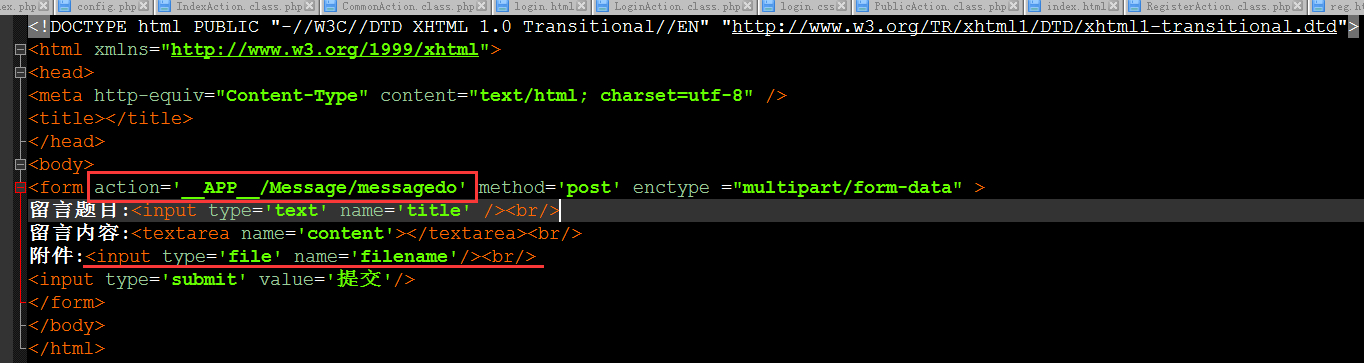
  
在TP手册当中会有说明；  
非常说明：  
这里的UserModel.class.php是我们新建的 所以当我们要实例化我们自己新建的一个类的时候我们采用的方式是使用D方法来实例化；当我们在使用create自动创建数据的时候那么这里会自动进行数据验证；

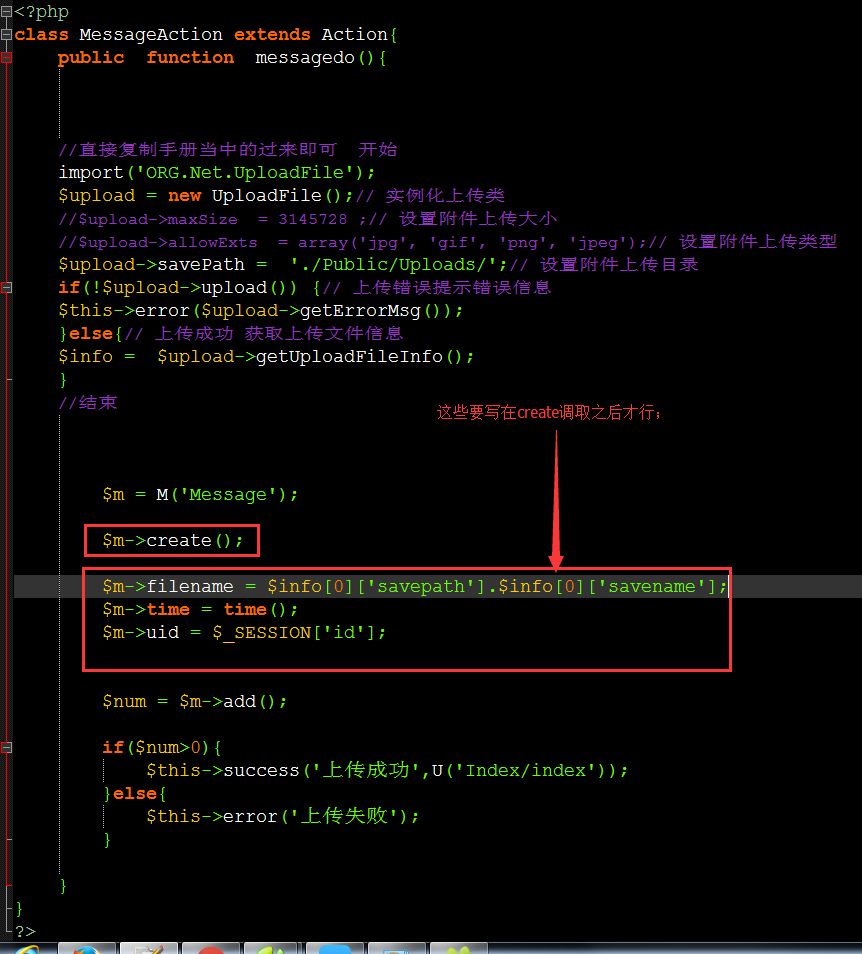
D方法就是来实例化我们自己创建的类的； 使用D方法来实例化自定义的类 但是如果这个类不存在的话那么就会默认的调取M方法来实例化这个不存在的类；

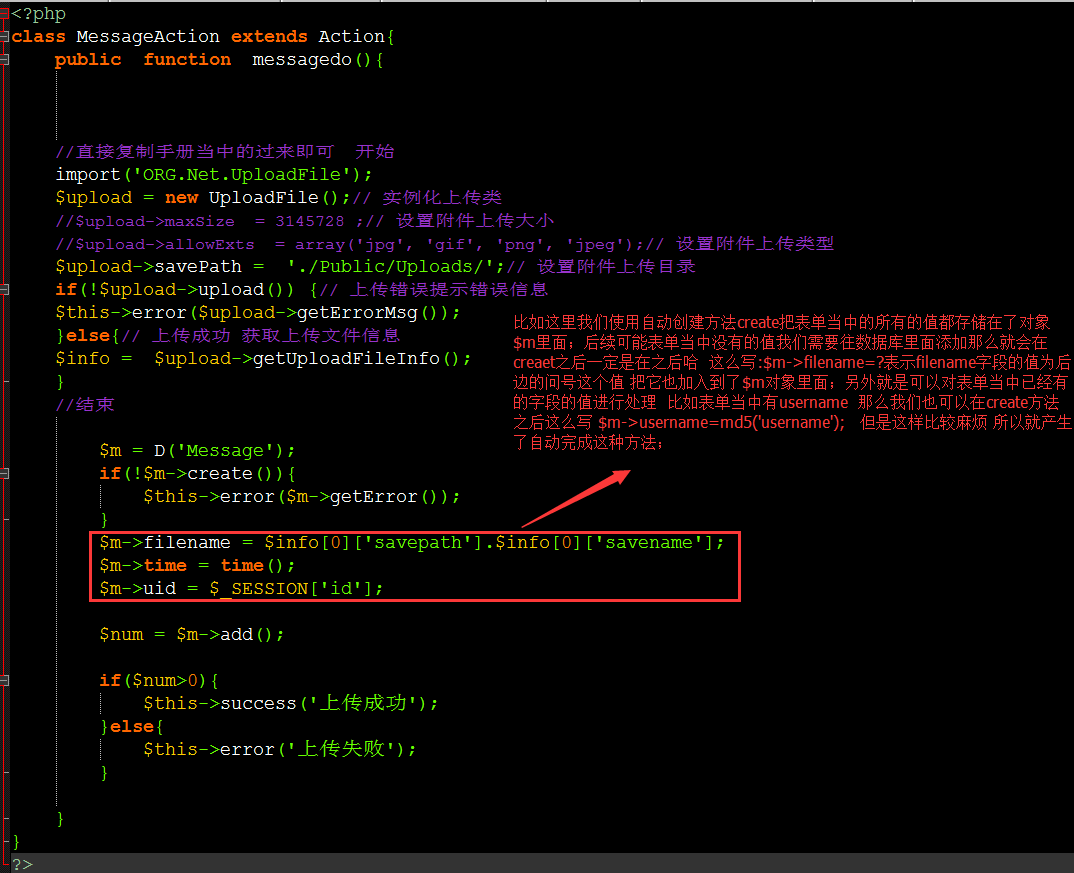
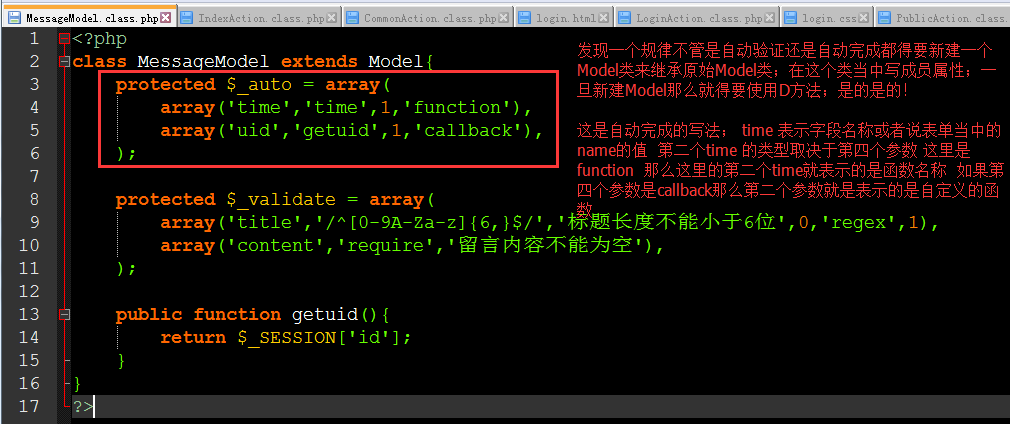
  
我们使用create自动创建的时候会有一个返回结果，自动创建成功的时候返回的是true 当自动创建失败的时候返回的是false 但是在自动创建的同时会进行自动验证，两个是同时进行的，所以当自动创建失败或者说验证失败的时候返回的都是false;抛出验证失败的方法是getError()方法；用实例化后的对象去调取这个getError（）方法既可以抛出错误提示也就是我们在自定义类userModel.class.php当中写的错误提示信息；

那么我们有没有办法使用自动验证来验证验证码的对错呢？  
在附加规则里面有一个callback方法验证；这个就是利用回调方法进行数据的验证的；  


上传问题：

  
在这里我们做了一个表单 提交留言的内容；前台界面是上边的代码形式的；后台代码是这样的：

  
我们把表单提交的位置到了Message类当中messagedo方法；在这个方法里面首先我们得要处理上传的文件，引入第三方类文件然后实例化，然后调取第三方类文件当中的upload方法；上传成功返回了上传资料的很多的详细的信息 上传失败跑出了上传失败的错误信息；  
然后我们利用自动创建的机制，前提是表单当中的name的值要和数据库当中的字段的名称保持一致；我们知道当我们实例化了M形成一个对象的时候 当我们调用create自动创建的时候那么就会把信息都加入到了$m当中去了；所以这里的$m->filename表示的是把对应的值也加入到了$m当中去；

自动完成：可以完成数据自动处理功能，用来处理默认值、数据过滤以及其他系统写入字段  
  


在thinkphp当中session默认是开启的；那么如何销毁session呢？

我们在php当中学习到的清除session的办法是：  
$\_SESSION = array( );

If ( isset ( $\_COOKIE[session\_name( )] ) ){

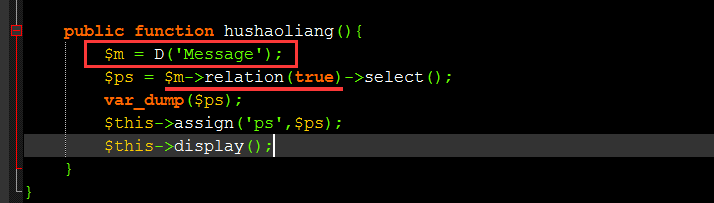
Setcookie(session\_name( ), ’’,time()-3600,’’/’ );  
}

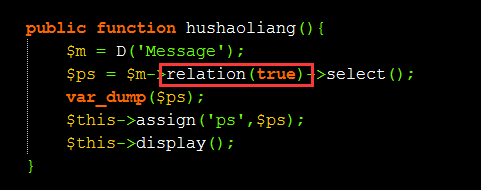
Session\_destory( );

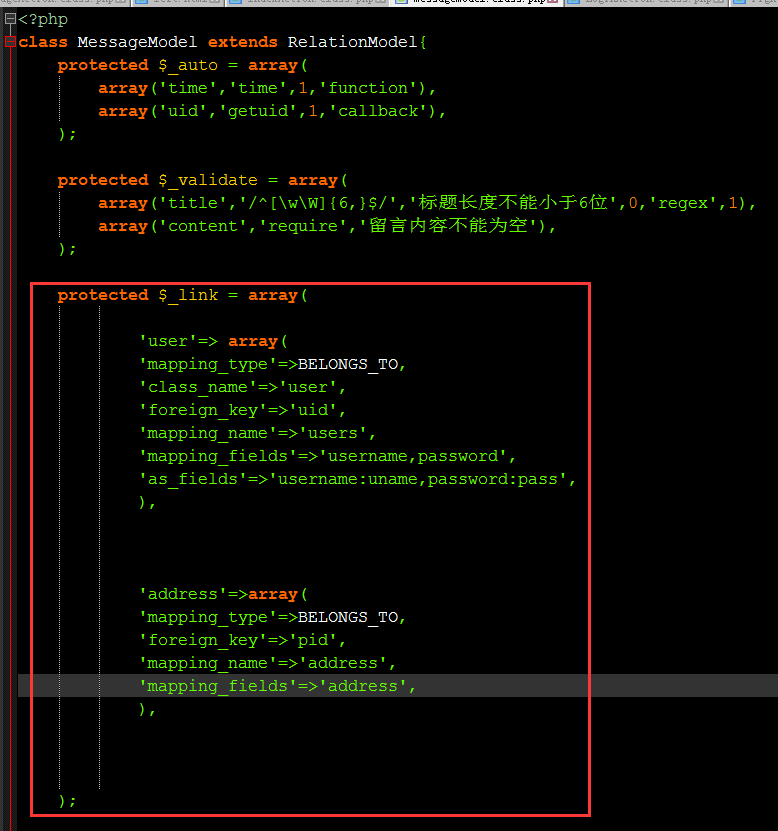
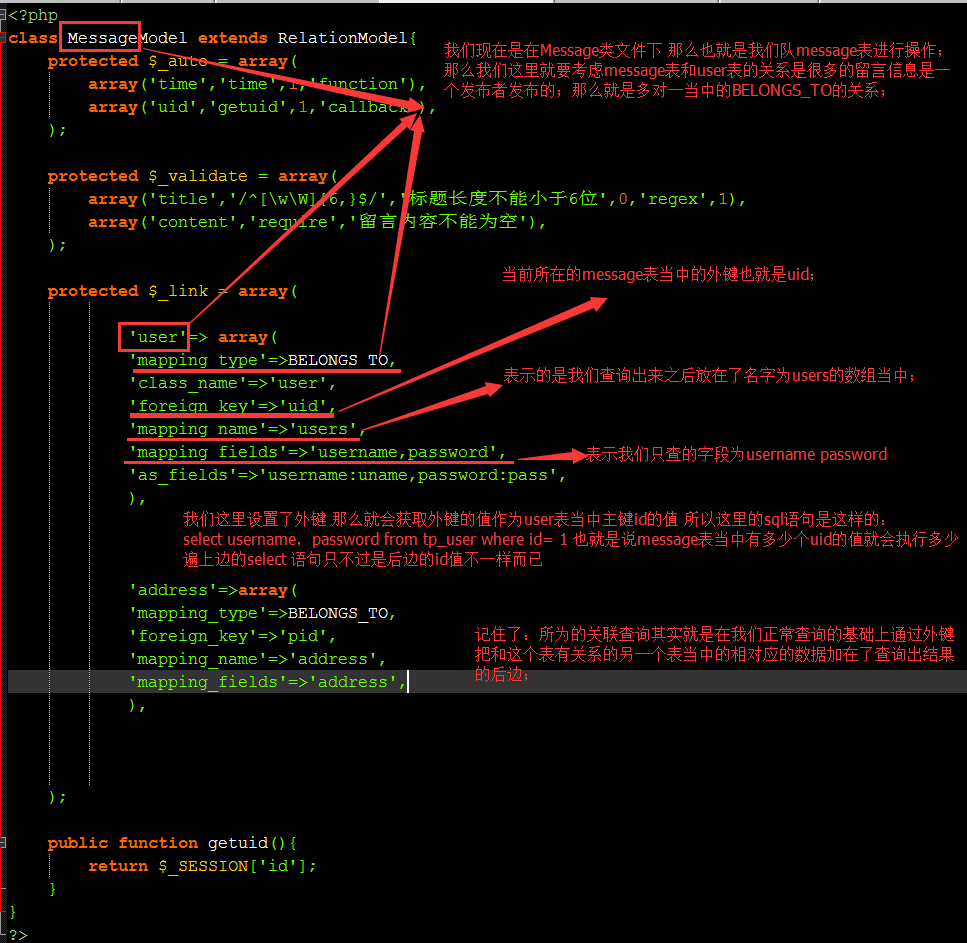
但是在thinkphp当中我们有更加简单的办法就是：  
session(null);//表示清除所有的session值；

Session(‘[destory]’);//清除服务器上所有的session文件；

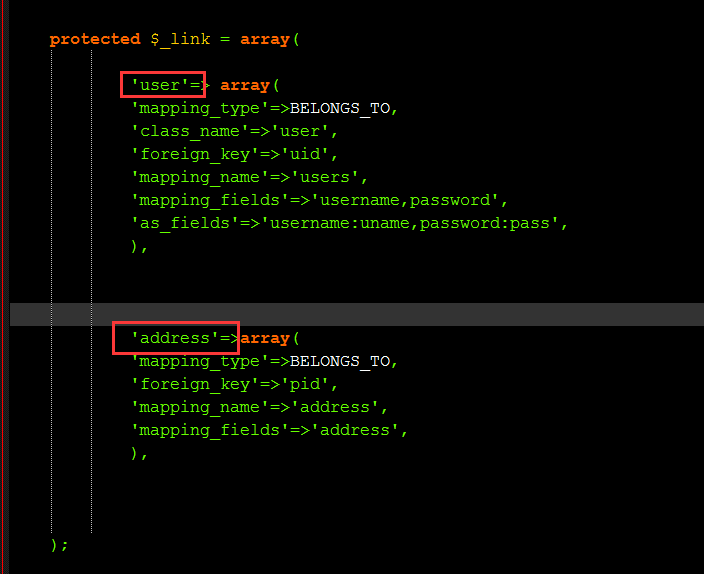
这两句就可以搞定了；

关联模型：（很复杂很重要哦）  
我们在做项目的时候比如我们把留言表当中所有的数据都差查出来了但是我们想获取每条留言的发布者是谁，在这个留言列表当中有个uid表示的是留言者的身份信息 这也就是外键；通过这个uid我们可以查找到user表当中的发布者姓名；如果按照正常的思维逻辑走的话我们会写很多的代码 但是在Thinkphp当中为我们提供了一种关联模型查询的方式；  


我们对数据进行操作首先要实例化Model 我们实例化自己写的Model用的是D；但是如果说我们要使用关联模型查询那么我们必须对这个Model类加上一个单词 也就是把所继承的Model类的名称改为：RelationModel：  
  
另外就是我们触发这个关联模型还需要在查询的时候使用relation进行触发:  
  
这哥relation里面放的是后边我们会在类文件当中写的关联的名称 以为关联不可能只能写一个 可以写多个或者更多：  
  
  

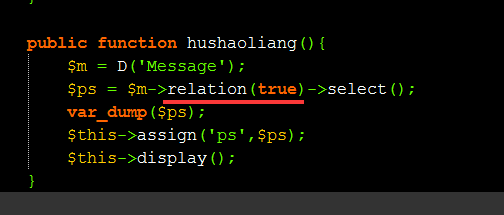

红色方框内的都是属于关联名称；  
模块控制层只需要写这么多；  
  
模型当中是这么写的:  
  
必须要写上protected $\_link = array( );在括号当中写关联实体；  


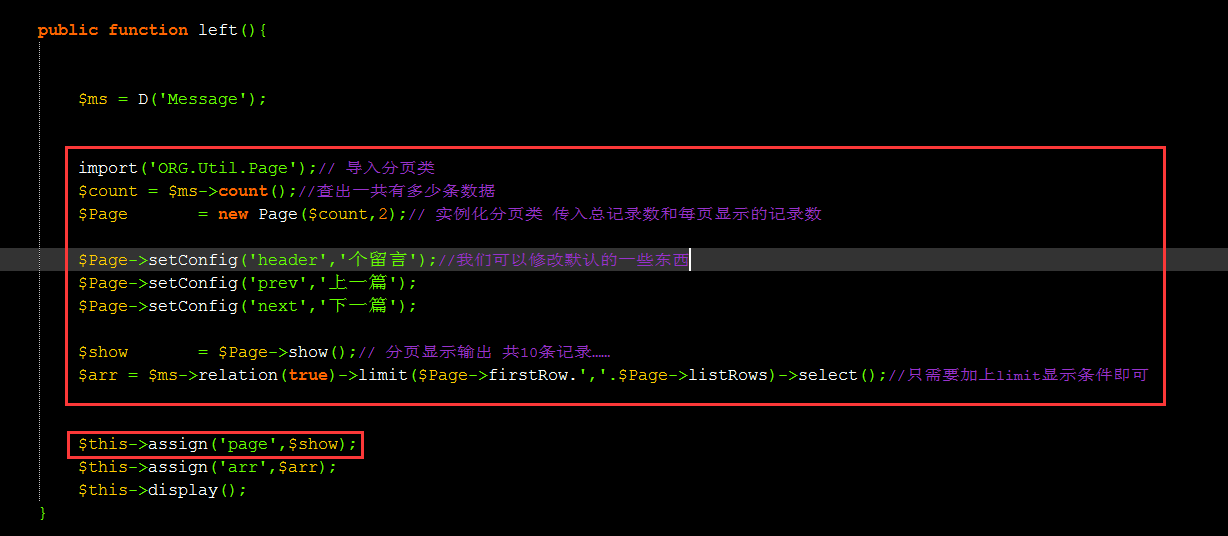
还要解释一个事情就是：



必须是表名称；

另外:

我们在触发的时候使用relation来触发 括号当中如果是true的话那么表示的就是关联1 和关联2当中我们设置的要查询出的字段的东西都能给查出来 但是我们不想要那么多的话那么我们可以在这里设置 在括号当中写关联的名称就行 比如在括号当中写user那么表示的就是触发user关联；  


分页显示：（直接参考手册即可 相当详细）  


THINKPHP到此结束