

ThinkPHP学习

作者: 王猛

Email: bietushiwo@gmail.com

指导: 燕十八

出自: 布尔教育--高端PHP工程师培训

注: 此文章可传播,可抄袭,传播请邮箱联系本人

框架的意义

- 减少重复劳动
- 便于团队配合
- 增强安全性
- 工作和面试都需要

为什么要学习Thinkphp

- 国内公司用的多
- 框架基本都是MVC框架,学一则通

框架学习的步骤

1. 安装部署
 - a. 下载解压 [readme INSTALL Demo 核心文件 等]
 - b. 引入
 - c. 生成项目
 - d. 创建新的模块 [网站前台/后台等部分,称为模块]
2. 控制器的使用
 - a. 路由器如何调用控制器 [URL与控制器/方法的关系]
 - b. 控制器文件叫什么?
 - c. 放在哪儿?
 - d. 继承谁?
3. 引入模板view
 - a. 模板放在哪儿?
 - b. 叫什么?
 - c. 和控制器有什么对应关系?
4. 按手册写个Model类
 - a. model类文件叫什么?
 - b. 放在哪儿?
 - c. 继承谁?
 - d. model类与表名的对应关系
 - e. 在controller里如何实例化
5. 如何配置
 - a. 配置文件放在哪儿?
 - b. 配置文件的优先级?
6. model操作
 - a. 创建数据
 - b. 更改数据
 - c. 删除数据
 - d. 查询数据

- 7. 模板语法
 - a. assign赋值
 - b. 变量输出
 - c. 数组循环
 - d. 三元运算
 - e. if判断
 - f. 模板包含
 - g. 导入css/js
 - h. 够了,不要让模板太重

第1章 框架部署

1.1下载

[TP官方网站]<http://www.thinkphp.cn/>

因为是中国人做的框架,所有全中文,注释明确,很容易就能找到下载地址.

目前最新是3.2.3 稳定版

建议:以后学习框架等等的新内容,就选用最新稳定版,不稳定版会有很多未修补的BUG,还在完善中,不建议学习最新测试版bata等等,最新测试版等大家对这个框架比较熟悉以后可以试着探索一下.

核心板core:只有框架的mvc核心功能

完整版full:有框架核心和实用类(上传,分页,验证码等)

这里我们用完整版,因为完整版不仅有mvc核心代码,还有我们常用的一些类库.可以直接看人家的源代码是如何完成的.

1.2引入

1. 解压文件
2. 得到ThinkPHP目录(只要这个就可以)
3. ThinkPHP可以放在任意目录(不一定非要WWW目录下)
4. 建立项目,如shop,cms,blog等
5. 项目目录下,创建index.php

index.php内容如下:

```
define('APP_PATH',__DIR__.'/'); //定义当前项目的地址
require('../ThinkPHP/ThinkPHP.php'); //引入TP源码文件
```

完成1.2的配置后,

在地址里面运行index.php即可自动生成项目

注意:Thinkphp中约定,目录必须以'/'结尾

1.3生成项目

看到如下界面,就配置成功了

```
:)
欢迎使用 ThinkPHP!
```

成功后,自动生成如下目录

Common --->库,类等可以放这儿

Home ---->controll model view都放这儿

Runtime ---->运行时生成的文件

1.4创建新模块(以Admin后台模块为例)

什么是模块:一个网站,常见的前台,admin是后台,二者的功能,css,图片等,都差的比较大,开发时,为了清晰,可以分为2个模块来管理

- 直接复制一份Home模块
- 并重命名为Admin
- 修改Admin/controller/indexController 的命名空间

```
namespace Home\Controller
```

修改为

```
namespace Admin\Controller;
```

地址栏访问 /index.php/Admin/Index/index

第2章 控制器的使用

2.1 URL如何调用控制器(重要)

以典型的URL(就是地址)为例 <http://shop.com/index.php/Home/Index/hello>
访问的Home模块下的IndexController的hello()方法

即index.php/模块名称/控制器/方法

那可不可以获取地址栏上的参数呢?

```
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller{
    public function index(){
        echo $_GET['uid'];
    }
}
```

如果需要传递其他参数,可以如下写

原来是XXX?uid=X

在thinkphp中封装了方法,如下

index.php/模块/控制器/方法/id/3

2.2 控制器放在哪里?

每个模块有自己的控制器,放在模块的Controller目录下

2.3 控制器文件叫什么?

文件名:控制器名Controller.class.php

例如:UserController.class.php

注意:单词的首字母大写[大驼峰规则]

2.4 控制器类叫什么?继承自谁?命名空间叫什么?

- 类名和文件名一致,本例叫UserController
- 继承自谁? Think\Controller
- 自身的空间什么?namespace和目录一致 例如:Home模块下的User控制器

```
namespace Home\Controller;
use Think\Controller;
```

```
class UserController extends Controller{
    public function niux(){
        echo '你真牛X';
    }
}
```

来,大家一起新建一个控制器,在Admin模块下新建:

1.先找到Admin模块下找到Controller层,并新建文件叫UserController.class.php

里面内容:

```
namespace Admin\Controller; \\此处应该是Admin了
use Think\Controller; \\用Think下的Controller是为了方便使用Controller这个类

class UserController extends Controller{
    public function ha(){
        echo "哈哈";
    }
}
```

2.访问:<http://localhost/shop/index.php/Admin/user/ha>

第3章 引入模版 view层

3.1 模版放在哪里?

放在模版的view目录下

并且,每一个控制器的模版,要在与控制器同名的目录下

思考:控制器的目录在哪叫什么来?

例如:index.php/Admin/User/ha

对应的模版就应该是:Admin/view/User/ha.html

3.2 叫什么?和控制器有什么对应关系?

模版名字和方法名字是对应的,一致的.

例如:地址栏:index.php/Admin/Uder/ha

对应的控制器是:Admin/UserController

对应的方法是:ha

那对应的模版就应该是:Admin/View/User/ha.html

代码如下:

1.在Admin目录下找到View目录下创建文件夹,此文件夹跟你控制器的名字一致,这里叫User,并且根据对应的方法创建对应的模版名字,这里叫ha.html

控制器里面代码如下:

```
namespace Admin\Controller;
use Think\Controller;

class UserController extends Controller{
    public function ha(){
        echo '你真牛X';
    }
}
```

2.如何显示出来是模版内容呢?

思考:smarty模版里面如何调用模版来?

```
namespace Admin\Controller;
use Think\Controller;
```

```
class UserController extends Controller{
    public function ha(){
        echo '你真牛';
        $this->display();//此处不需要写模版名称,对应文件创建文件夹,文件名一致就是为了方便调用,不需要写模版的名称
    }
}
```

第4章 Model层

4.1 Model在哪？

项目/模块/Model目录下

以本例子来看,应该是Home模块下的Model

/Home/Model/目录下

4.2 model类文件叫什么？

模型名:XxModel.class.php

4.3 model类与表名的对应关系

一般model名称与表名table相同,回忆:什么是table表?

例如:GoodsModel.class.php 则对应应该操作的表示goods表

注意:虽然Model类首字母大写的,但按照平时操作,表名都是小写

4.4 继承谁？

先明白为什么要继承...

```
namespace Home\Model
use Think\Model

class XxModel extends Model{

}
```

4.5 在controller里实现实例化

按照mvc的架构,c是业务逻辑,m是处理数据的,v是负责显示的.

我的流程就是:把(M层)数据里面的信息拿出来或者添加到数据库,然后到c层做业务逻辑处理,完了再到v层显示出来.

上面讲到,C层的显示模版的操作实际上是在C层写的,v只负责显示.那m和c层又是操作的?

用户只能访问的是c层,C层负责调用m层,m层去数据库拿数据或者往里面填数据,完了以后把数据扔给c层做处理,c层再把数据扔给v层显示.所以,你必须利用c层去调用m层

例如:

找到模块下,Home为例,Home模块下有个Model文件夹,在里面新建

注意上面的命名规则

GoodsModel.class.php

```
namespace Home\Model;
use Think\Model;

class GoodsModel extends Model{
    public function nini(){
        echo '就当这里是数据库出来的数据';
    }
}
```

按照上面讲的,我们需要在控制器(c)层去使用这里的数据,那一定需要new了

```
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller{
    public function index(){
        //echo 123;
        //echo $_GET['uid'];
    }

    public function wox(){
        $model = new \Home\Model\GoodsModel();
        var_dump($model);
    }
}
```

发现报错了.哭的脸,页面错误!请稍后再试~

我们忘了一件事情,框架里面一般都会开启报错模式,是让你能看到错误信息,知道错误在哪里.

找到咱刚开始解压出来的饿完整的框架包,里面也有一个index.php,打开瞅一眼看看.

```
// 检测PHP环境
if(version_compare(PHP_VERSION,'5.3.0','<')) die('require PHP > 5.3.0 !');

// 开启调试模式 建议开发阶段开启 部署阶段注释或者设为false
define('APP_DEBUG',True);

// 定义应用目录
define('APP_PATH','./Application/');

// 引入ThinkPHP入口文件
require './ThinkPHP/ThinkPHP.php';
```

注意:把开启调试模式放在配置的开头.

发现还是报错,但报错信息有了.依然是哭脸,无法加载数据库驱动:Think\Db\Driver\

看一下,如果不new发现方法正常运行,一旦new就报错,难道不能再方面里面new类?还是在new完了以后做的事情出错了?

如果不是,应该报错信息说的是,new Model不成功.这说明,你new成功了.new完了以后框架内部又做了一些事情,这里面报错了.提示:面向过程里面你要连接数据库需要先干什么?是不是需要地址用户名密码啊?说明new完了以后,人家找不到数据库啊

马上完成作业:自己新建Model把错误打印出来

第5章 配置

5.1 配置文件在哪?

TP框架自带的配置文件:

/ThinkPHP/Conf/convention.php

放在哪里呢?

把相关的数据库连接的内容放在我们项目中/Common/Conf/config.php

```
/* 数据库设置 */
'DB_TYPE'      => 'mysql',      // 数据库类型
'DB_HOST'      => 'localhost',  // 服务器地址
'DB_NAME'      => 'tp',         // 数据库名
'DB_USER'      => 'root',       // 用户名
'DB_PWD'       => 'root',       // 密码
'DB_PORT'      => '3306',       // 端口
```

```
'DB_PREFIX' => '', // 数据库表前缀,老项目会用到
```

我们还没有数据库,新建一个,叫tp库,表要跟Model方法名同名,叫goods,随便再来几个字段,这里用,id,name,age

试一下:<http://127.0.0.1/shop/index.php/home/index/wox>

5.2 文件的优先级

注意:配置文件选项都是一样的,越小的配置文件优先级越高.

有的项目,分数据库,有一些项目前端用的数据库可能并不是mysql,后端用的mysql,这样就出现了不同的数据库,在每一个模块中,也有配置选项,当我们设置了不同模块的数据库的时候,整体的项目配置文件就会被模块中的配置覆盖掉.

调用选项:

项目的入口页面,一般都是index.php

所有文件要读取,首先要进入index.php

其中常用的还有:

```
define('APP_DEBUG',true); //开启debug模式

//记录日志
'LOG_RECORD' => true, //系统日志在记录的时候需要开启debug调试模式,如果debug模式没有开启,日志并不记录.
```

日志在哪?

/项目(shop)/Runtime/Logs/ 下分模块记录的日志

第6章 Model操作

6.1 建立栏目表用于联系

```
CREATE TABLE cat (
  cat_id int(11) NOT NULL AUTO_INCREMENT,
  cat_name varchar(20) NOT NULL DEFAULT '',
  intro varchar(100) NOT NULL DEFAULT '',
  parent_id int(11) NOT NULL DEFAULT 0,
  PRIMARY KEY ( cat_id )
) ENGINE=MyISAM CHARSET=utf8;
```

6.2 新增数据

对于TP的Model,支持2种风格的操作

对应的表名应该跟对应的Model新建相应的class文件

这里使用cat表

1)在Home模块下新建CatModel.class.php

```
namespace Home\Model;
use Think\Model;
class CatModel extends Model{

}
```

2)在Controller层IndexController.class.php新建方法

```
public function adds(){
  $catModel = new \Home\Model\CatModel();
  $data = array('cat_name'=>'小刘','intro'=>'我也是普通人嘛','parent_id'=>0);
  $catModel->add($data);
}
```

```
}
```

3).访问/shop/index.php/home/index/adds 如果啥玩意没有,看数据库有没有信息就知道添加是否成功

以上注意:组织数据的时候是以数组的方式进行添加,数组的key就是表中字段的名字,而value就是对应的字段值

概况如下:

1:以上是面向过程风格

2: 面向对象风格:

\$model->属性名 = 属性值

\$model->add();

例:

```
$catModel->cat_name = 'xibeiwang';  
$catModel->intro = 'guofei';  
$catModel->parent_id = 1;  
$catModel->add();
```

思考:为什么后面这个不需要传递参数?

__get(\$a) __set(\$a,\$b) 不可见的时候触发这个方法.ORM方式:对象关系映射,ORM是通过使用描述对象和数据库之间映射的元数据,将程序中的对象自动持久化到关系数据库中

注意:成功后返回的是主键的值

6.3 查询数据

注意:查询时find(),select(),分别查1行,和多行时用

- 按主键查询:\$model->find(值);
- 手动传参查询:\$model->where('id>100')->find();

操作: 以新建为例

1).在Controller里面新建一个UserController.class.php

```
namespace Home\Controller;  
use Think\Controller;  
  
class UserController extends Controller{  
    public function cha(){  
        //新方法 tp封装的方法  
        $catModel = D('Cat');  
        var_dump($catModel->find('2'));  
    }  
}
```

2)select查询

```
namespace Home\Controller;  
use Think\Controller;  
  
class UserController extends Controller{  
    public function cha(){  
        //新方法 tp封装的方法  
        $catModel = D('Cat');  
        //var_dump($catModel->find('2'));  
        //var_dump($catModel->select('2'));  
        //var_dump($catModel->where('cat_id=3')->select());  
    }  
}
```

6.4 更改数据

想修改数据,需要先找哪行数据,当然不需要查出来也是可以的
用的是save()方法

```
namespace Home\Controller;
use Think\Controller;

class UserController extends Controller{
    public function cha(){
        //新方法 tp封装的方法
        $catModel = D('Cat');
        //var_dump($catModel->find('2'));
        //var_dump($catModel->select('2'));
        //var_dump($catModel->where('cat_id=3')->select());
    }
    public function gai(){
        $catModel = D('Cat');

        //面向对象的风格
        $catModel = find('2');
        $catModel -> cat_name = 'xibeiwang';
        var_dump($catModel->save()); //返回受影响行数
        //不查询直接修改也可以

        //其实就是面向过程化的风格
        $catModel->where('cat_id=1')->save(array('cat_name'=>'haoshuang'));
    }
}
```

6.5 删除数据

删除的方法类似,用的是delete()方法

```
var_dump($model->delete(4)); //按主键来删除 where cat_id=4
var_dump($model->delete('3,4,5,6')); // where cat_id in (3,4,5,6);
var_dump($model->where('cat_id=2')->delete()); // where cat_id=2;
```

```
namespace Home\Controller;
use Think\Controller;

class UserController extends Controller{
    public function cha(){
        //新方法 tp封装的方法
        $catModel = D('Cat');
        //var_dump($catModel->find('2'));
        //var_dump($catModel->select('2'));
        //var_dump($catModel->where('cat_id=3')->select('2'));
    }
    public function gai(){
        $catModel = D('Cat');

        //面向对象的风格
        //$catModel = find('2');
        //$catModel -> cat_name = 'xibeiwang';
        //var_dump($catModel->save()); //返回受影响行数
        //不查询直接修改也可以

        //其实就是面向过程化的风格
        //$catModel->where('cat_id=1')->save(array('cat_name'=>'haoshuang'));
    }

    public function shan(){
        $catModel = D('Cat');
        echo $catModel->where('cat_id=1')->delete();
    }
}
```

第7章 模版操作

7.1 assign赋值

模版的输出跟smarty一样的

```
$this->assign('title','标题');
$this->assign('content',$content);
$this->display();
```

7.2 变量输出

```
<title>{$title}</title>
```

注意:TP的模版标签和smarty有一些地方稍有不同

TP在控制结构/循环/if/else时,用的是<>

而具体输出的变量用的是{\$标签}

举例: 这里用上面已经写好的方法

1)

```
namespace Home\Controller;
use Think\Controller;

class UserController extends Controller{
    public function cha(){
        //新方法 tp封装的方法
        $catModel = D('Cat');
        //var_dump($catModel->find('2'));
        //var_dump($catModel->select('2'));
        //var_dump($catModel->where('cat_id=3')->select('2'));
    }
    public function gai(){
        $catModel = D('Cat');

        //面向对象的风格
        //$catModel = find('2');
        //$catModel -> cat_name = 'xibeiwang';
        //var_dump($catModel->save());//返回受影响行数
        //不查询直接修改也可以

        //其实就是面向过程化的风格
        //$catModel->where('cat_id=1')->save(array('cat_name'=>'haoshuang'));
    }

    public function shan(){
        //$catModel = D('Cat');
        //echo $catModel->where('cat_id=1')->delete();
        $this->assign('title','我是标题');
        $this->display();
    }
}
```

2)在模版中显示

/Home/view/User/shan.html

```
....
<h1>{$title}</h1>
....
```

7.3 if/else标签

在手册里面 模版/内置标签/比较标签

注意:TP里面的判断跟smarty不同的是,需要使用标签具体的英文缩写

```
<if condition="$rand egt 5">及格<else/>不及格</if>
```

内容

</比较标签>

1)

```
namespace Home\Controller;
use Think\Controller;

class UserController extends Controller{
    public function cha(){
        //新方法 tp封装的方法
        $catModel = D('Cat');
        //var_dump($catModel->find('2'));
        //var_dump($catModel->select('2'));
        //var_dump($catModel->where('cat_id=3')->select('2'));
    }
    public function gai(){
        $catModel = D('Cat');

        //面向对象的风格
        //$catModel = find('2');
        //$catModel -> cat_name = 'xibeiwang';
        //var_dump($catModel->save());//返回受影响行数
        //不查询直接修改也可以

        //其实就是面向过程化的风格
        //$catModel->where('cat_id=1')->save(array('cat_name'=>'haoshuang'));
    }

    public function shan(){
        //$catModel = D('Cat');
        //echo $catModel->where('cat_id=1')->delete();
        //$this->assign('title','我是标题');
        $rand = mt_rand(1,9);
        $this->assign('rand',$rand);
        $this->display();
    }
}
```

2)在模版中判断

```
.....
<if condition="$data gt 5">
    你及格了
<else/>
    没及格
</if>
.....
```

7.4 三元运算

手册:/模版/三元运算

```
namespace Home\Controller;
use Think\Controller;

class UserController extends Controller{
```

```

public function cha(){
    //新方法 tp封装的方法
    $catModel = D('Cat');
    //var_dump($catModel->find('2'));
    //var_dump($catModel->select('2'));
    //var_dump($catModel->where('cat_id=3')->select('2'));
}

public function gai(){
    $catModel = D('Cat');

    //面向对象的风格
    //$catModel = find('2');
    //$catModel -> cat_name = 'xibeiwang';
    //var_dump($catModel->save());//返回受影响行数
    //不查询直接修改也可以

    //其实就是面向过程化的风格
    //$catModel->where('cat_id=1')->save(array('cat_name'=>'haoshuang'));
}

public function shan(){
    //$catModel = D('Cat');
    //echo $catModel->where('cat_id=1')->delete();
    //$this->assign('title','我是标题');
    $rand = mt_rand(0,1);
    $this->assign('rand',$rand);
    $this->display();
}
}

```

2)模版中

```

....
{$rand?'对了':'错了'}
....

```

7.5 数组循环

大概如下:引导了两次手册,你应该学会用手册,自己查,以下给你列举到下面

```

<foreach name="bests" item="g">
<li>
    {$g.goods_name}
</li>
</foreach>

```

```

namespace Home\Controller;
use Think\Controller;

class UserController extends Controller{
    public function cha(){
        //新方法 tp封装的方法
        $catModel = D('Cat');
        //var_dump($catModel->find('2'));
        //var_dump($catModel->select('2'));
        //var_dump($catModel->where('cat_id=3')->select('2'));
    }
    public function gai(){
        $catModel = D('Cat');

        //面向对象的风格
        //$catModel = find('2');
        //$catModel -> cat_name = 'xibeiwang';
        //var_dump($catModel->save());//返回受影响行数
        //不查询直接修改也可以
    }
}

```

```
//其实就是面向过程化的风格
// $catModel->where('cat_id=1')->save(array('cat_name'=>'haoshuang'));
}

public function shan(){
    // $catModel = D('Cat');
    // echo $catModel->where('cat_id=1')->delete();
    // $this->assign('title','我是标题');
    // $rand = mt_rand(0,1);
    $rand = ['找','钱','孙','子'];//5.5以后可以使用这种方法
    $this->assign('rand',$rand);
    $this->display();
}
}
```

2)模板中

```
....
<foreach name="rand" item="v"> // 此处的name就是传过来的标签名,v代表的是值
    {$v} // 当然,数据库里面出来的时候后面可以加.固定的变量
</foreach>
....
```

定义变量k怎么办?只需要在输出的时候加{\$key}即可自动输出(参考手册)

7.6 impor导入CSS,JS文件

impor标签,以APP/Public为起点,导入css,js文件

```
<import type="css" file="a.b.c" />
```

则导入

```
APP/Public/a/b/c.css //APP指代项目跟目录,新建Public/a/b目录
```

实战如下:

shan.html

```
....
</head>
<import tyoe="css" file="a.b.c" />
....
```

Public/a/b/c.css 里面写css样式,会自动加载到刚才的路径文件中

7.7 load导入css,js文件

实例:

```
<load href="__PUBLIC__/Js/XXX.js" />
```

导入的是APP/Public/Js/xxx.js

注: 7.6,7.7中的APP均指项目目录

实战:

```
....
</head>
<load href="__PUBLIC__/Js/a.js" />
....
```

Public/Js/a.js

7.8 模版包含

一般在模块下的view中,建立Public ,里面放置公共模板
如:

```
/view/Public/  
header.html  
footer.html
```

如何包含?

```
<include file="Public/模版名(不含后缀)" />  
例如:包含Home/View/Public/header.html;  
<include file="Public/hader" />
```

7.9 不要让模版做太多的工作

功能:输出数据,而非处理数据
应该由控制器来处理

第8章 全局函数

8.1 M函数

Thinkphp的Model父类,封装的功能非常多,增删改查操作都具备
一些表,比如说留言表,comment

```
class CommentModel extends Model{  
  
}
```

M('comment')的作用就是找到父类的Model类中的对象,
且把Model的表名---指向--->comment
D()方法和M()方法的就别就在于,一个需要new 继承以后才可以使用D()方法,而M()方法是父类里面直接可以用的方法.

D方法可以自动检测模型类, 如果存在自定义的模型类, 则实例化自定义模型类, 如果不存在, 则会实例化Model基类, 同时对于已实例化过的模型, 不会重复去实例化。

M方法 就如一台刚装好操作系统的电脑, 只能使用一些基本功能; 而 D方法 就如在装好的系统上再安装了一些如 Office、QQ 等应用软件, 功能更加强大, 同时整个电脑运行速度也变慢了。

8.2 D函数

如上

8.3 D函数跨模块调用Model

例如:在Admin/Controller/index/index方法中
我想调用Home下面model下面的catModel,如何调用?

```
....  
public function aa(){  
    new \Home\Model\CatModel();  
    D('Home/Cat');  
}  
....
```

8.4 U函数

U函数按系统设定的URL规则,生成URL

主要用途:在我们删除/修改/传值的时候使用地址进行跳转的时候使用的函数
例如:

index.php?Home/Index/index

index.php?Home-Index-index.html

在模板中的URL,通过U函数生成,可以避免后期URL规则变化,
带来的修改工作.

在控制器中,可以这样来写:

\$url = U('模块/控制器/方法',参数数组或字符串);

例:

```
U('Home/Index/index',array('id'=>3));
U('Home/Index/index','id=5&page=3');
```

注意:在模板中调用时,标签中加":

```
{:U('Home/Index/index','id=5&page=3')}
```

模板中如下例句:

```
<foreach name="cats" item="c">
<tr>
<td>{$c.cat_id}</td>
<td>{$c.catname}</td>
<td>{$c.parent_id}</td>
<td>
栏目<a href="{:U('Home/Index/index' , array('cat_id'=>{$c[cat_id]})}">编辑</a>
</td>
</tr>
</foreach>
```

8.5 C函数

读取和设置配置选项的函数

可以动态的去设置你的配置选项

例如:

Home/index.php/User/shan

我们的config.php中设置了公共的属性

```
....
//echo C('DB_TYPE');
//在config中进行自定义属性设置,也可以获取到
echo C('laowang');
....
```

还可以自定义进行参数设置

```
....
C('laowang','zhengchang');
echo C('laowang');
....
```

注意:它改变的并不是文件信息,而是临时的运行时加载到内存的环境

8.6 I函数

不管是GET传参还是POST都可以使用I函数进行接收

防止XSS攻击和SQL注入所做的过滤函数

1).\$_GET

例如:

Home/index.php/User/shan

```
....  
echo I('id');  
....
```

form表单中

```
<a href="{:U('Home/Index/index',array('id'=>123))}"
```

2).\$_POST

Home/View/User/shan.html

新建form表单

```
....  
<form action="{:U('Home/User/shan')}" method="post">  
  <input type="text" name="name" value="">  
  <input type="submit" value="提交">  
</form>  
....
```

Controller层shan

```
....  
if(!$POST){  
  $this->display();  
  var_dump(I('name'));  
  //如果要传递所有的值怎么办?get.值/post.值  
  var_dump(I('post.'));  
}  
....
```

第9章 实战商城前台页面

原来的shop不要了,新建一个shop,把原来的shop换个名字备注起来,练习的时候使用shop0里面的代码

9.1 配置虚拟主机

```
host文件 配置  
C:\Windows\System32\drivers\etc  
127.0.0.1 shop.com
```

apache vhost.conf

```
<VirtualHost *:80>  
  DocumentRoot "D:/xampp/hotdocs/shop"  
  ServerName shop.com  
</VirtualHost>
```

9.2 引入模版

在Home/view下建立目录如下:

```
/Home/view  
/User  
/Index
```


分别把:

```
Cat    ----  cat.html
        catlist.html

Goods ----  checkout.html
        done.html
        goods.html

Index ----  index.html

User   ----  login.html
        msg.html
        reg.html
```

9.3写对应的控制器

```
class IndexController extends Controller {
    public function index(){
        //不加参数,默认找index.html做模版
        $this->display();
    }
    public function cat() {
        //其实也可以加模版名自,但不要加".html"后缀
        $this->display('category');
    }
    public function goods(){
        $this->display();
    }
}
```

9.4 存放CSS/JS/images资源

网站的css/js/图片等静态资源,TP一般是放在项目/Public目录下

如果没有,可手动创建,把模块下的资源拷进来

如果模块不止一个,还可以在Public再建子目录

```
/shop
/Public
/Home
/Admin
```

把模版中的资源放在相应的目录中

9.5 修改模板中资源路径

项目的/Public目录地址,如何表示?

可以用TP提供的"模板常量"来书写, **PUBLIC**

在本例中,

css--替换为--> `__PUBLIC__ /Home/css`

images--替换为--> `__PUBLIC__ /Home/images`

js--替换为--> `__PUBLIC__ /Home/js`

例:index.html

```
<link href="./css/style.css" rel="stylesheet" type="text/css" />
```

改为:

```
<link href="__PUBLIC__/Home/css/style.css" rel="stylesheet" type="text/css" />
```

css--替换为--> **PUBLIC**/Admin/css

images--替换为--> **PUBLIC**/Admin/images

js--替换为--> **PUBLIC**/Admin/js

第10章 后台搭建

在Admin/view下建立目录如下:

```
Cat ---- cateadd.html
        catelist.html

Goods ---- goodsadd.html
         goodslist.html

Index ---- drag.html
         index.html
         left.html
         main.html
         top.html
```

后面学员自己完成,仿照前台的方法,把后台的CSSjs等存放好,写好对应的方法进行显示

例如:

/Admin/Controller/IndexController

```
namespace Admin\Controller;
use \Think\Controller;
class IndexController extends Controller {
    public function index(){
        $this->display();
    }
}
```

10.2 栏目发布

1).先得有数据库.利用phpmyadmin导入数据库

2).配置数据库信息config.php

```
// '配置项' => '配置值'
'DB_TYPE'           => 'mysql',      // 数据库类型
'DB_HOST'           => '127.0.0.1',  // 服务器地址
'DB_NAME'           => 'shop',       // 数据库名
'DB_USER'           => 'root',       // 用户名
'DB_PWD'            => '',           // 密码
'LOG_RECORD'        => true,         // 系统日志在记录的时候需要开启debug调试模式,如果debug模式没有开启,日志并不记录.
```

访问对应的模块

<http://127.0.0.1/shop/index.php/Admin/cat/cateAdd>

发现多了很多东西,删除不需要的东西

3).修改对应模版中的值,回忆blog如何做的,TP又应该怎么办?

里面包括 action/name值等

```
action="{:U('Admin/Cat/Cateadd')}"
```

name对应数据库的字段进行修改

4).在逻辑层写相应的方法进行数据添加

```
public function add() {
```

```

if(!IS_POST) {
    $this->display('cateadd');
    exit;
}else{
    //var_dump($_POST);
    //没有Model层,去model一下
    $catModel = D('Cat');
    $catModel->add($_POST);
}
}
}

```

10.3 栏目列表

1).对应的方法

```

public function catelist(){
    $catModel = D('Cat');
    $catlist = $catModel ->select();
    //var_dump($catlist);
    exit;
    $this->assing('list',$catlist);
    $this->display();
}

```

2).模板中对应的遍历

catelist.html

```

<foreach name="list" item="ls">
    ....
    <span><a href="#">{$ls['cat_name']}</a></span>
    ....
</foreach>

```

10.3 栏目编辑和删除

catlist.html

1).先编辑对应的模版

```

....
<a href="{:U('Admin/Cat/catedit',array('cat_id'=>$ls['cat_id']))}">编辑</a>
....

```

2).到对应的方法中执行修改操作

```

....
public function catedit(){
    $cat_id = I('cat_id');
    $catModel = D('Cat');
    $catinfo = $catModel->find($cat_id);
    //var_dump($catinfo);
    $this->assign('info',$catinfo);
    $this->display();
}
....

```

3).没有对应的模版,复制一份cateadd.html改名为catedit.html

并修改分类名称:

```

....
//提交到"{:U('Admin/Cat/Catedit')}"
<input type="text" name="cat_name" value="{ $info['cat_name']}"
<textarea name="intro">{$info['intro']}</textarea>

```

```
.....
```

4).POST过来的信息并没有ID值,利用隐藏域

```
.....
<input type='hidden' name='cat_id' value="{ $info['cat_id']}" />
.....
```

5).执行修改和删除操作

```
public function catedit(){
    $catModel = D('Cat');//拿到外边来,因为else也要用
    if(!IS_POST){
        $cat_id = I('cat_id');
        $catinfo = $catModel->find($cat_id);
        //var_dump($catinfo);
        $this->assign('info',$catinfo);
        $this->display();
    }else{
        $catModel->where('cat_id='.$_POST['cat_id'])->save($_POST);
    }

}

public function del(){
    //记住先去模板中修改点击的地址
    $catModel = D('Cat');
    $catModel->delete(I("get.cat_id"));
    //$this->redirect('Admin/Cat/catelist');
    $this->success('成功','',5);
}
```

10.4 商品的发布

Model在add()或save前,属性[对应列]得先有值.

如何让他有值?

// 办法1 手动赋值

\$model->属性1 = 值1;

\$model->属性2 = 值2;

// 办法2 批量赋值

\$model->create(\$arr);

// 办法3 直接create()

create()会自动分析POST数据,并赋值到Model的"数据对象"上,
再调用add(),save()就可以添加或修改

10.5 自动验证

定义好各字段的验证规则.

如长度在6-16之间,或必须是email等等

在TP中,在model中写验证规则,即\$_validate 属性

规则为:

```
array(
    array(验证字段1,验证规则,错误提示,[验证条件,附加规则,验证时间]),
    .....
);
```

实战:

1).修改goodsadd.html中需要商品添加的模版

goodsadd.html

```
....  
action="{:U('Admin/goods/goodsadd')}"  
....
```

2).在对应的goodsController.class.php中写相应的方法

goodsController.class.php

```
namespace Admin\Controller;  
use Think\Controller;  
  
class GoodsController extends Controller{  
    public function goodsAdd(){  
        if(!IS_POST){  
            $this->display();  
        }else{  
            var_dump($_POST);  
        }  
    }  
}
```

3).自动验证需要用model去调用一个create方法

需要在Model层新建GoodsModel.class.php

```
namespace Admin\Model;  
use Think\Model;  
class GoodsModel extends Model{  
}
```

4).使用model出来的create方法

参考手册里面自动验证 模型/自动验证

```
namespace Admin\Controller;  
use Think\Controller;  
  
class GoodsController extends Controller{  
    public function goodsAdd(){  
        if(!IS_POST){  
            $this->display();  
        }else{  
            //var_dump($_POST);  
            $goodsModel = D('Goods');  
            if(!$goodsModel->create()){  
                echo $goodsModel->getError();  
                exit();  
            }else{  
                echo "成功";  
            }  
        }  
    }  
}
```

在model中填写规则 GoodsModel.class.php

```
....  
protected $_validate = array(  
    array('goods_name','3,10','必须是3-6位字符串',1,'length',3),  
    array('goods_sn','','数据库已经存在.',1,'unique',3),  
);  
....
```

5).执行添加

```
namespace Admin\Controller;
use Think\Controller;

class GoodsController extends Controller{
    public function goodsAdd(){
        if(!IS_POST){
            $this->display();
        }else{
            //var_dump($_POST);
            $goodsModel = D('Goods');
            if(!$goodsModel->create()){
                echo $goodsModel->getError();
                exit();
            }else{
                $goodsModel->add();
            }
        }
    }
}
```

10.6 自动填充

像有一些字段,不给你值,希望你有一个默认值,比如说注册时间,添加时间等等

给Model定义__auto属性,即可完成自动填充

```
array(
    array(完成字段1,完成规则,[完成条件,附加条件]),
    array(完成字段1,完成规则,[完成条件,附加条件])
    .....
);
```

对应手册的信息,如果是对商品Model设定填充规则

```
protected $_auto = array(
    array('add_time','time',1,'function'),
    array('last_update','time',2,'function'),
);
```

10.7 自动过滤

对于某些片段,是不允许后期修改的,如身份证号,商品货号等.

但是对于TP的"自动完成"来说,只要是表中合法的字段,都将接收,并自动帮我们更新--这是非常危险的.

TP框架里面可以做到自动过滤到不属于表中的字段,这是框架做的安全方面的防范.但是像身份证不变,但人家想修改一下手机号,拼凑一下身份证,一起发到数据库,可能就修改了.

goodsmodel.class.php

```
....
protected $insertFields = 'goods_id,cat_id,goods_name,goods_number,
goods_weight,shop_price,goods_desc,
goods_brief,ori_img,goods_img,thumb_img,is_best,
is_new,is_hot,is_sale,add_time
....
```

10.8 商品列表页

1)控制器做列表的相关操作

```

class GoodsController extends Controller {
public function lists() {
    // 查询商品
    $model = new \Home\Model\GoodsModel();
    $goods = $model->field('goods_id,goods_name,goods_sn,shop_price,is_on_sale,is_best,is_new,is_hot,goods_number')->order('goods_id');
    // print_r($goods);
    $this->assign('goods',$goods);
    $this->display();
}
}

```

2). 模板中做相应的模版操作

goodslist.html

```

....
<foreach name='goods' item='g'>
<tr>
    <td><input type="checkbox" name="checkboxes[]" value="32" />{$g.goods_id}</td>

    <td class="first-cell" style=""><span>{$g.goods_name}</span></td>
    <td><span>{$g.goods_sn}</span></td>
    <td align="right"><span>{$g.shop_price}</span></td>
    <td align="center"></td>
    <td align="center"></td>
    <td align="center"></td>
    <td align="center"></td>

    <td align="center"><span>100</span></td>
    <td align="right"><span>1</span></td>
    <td align="center">
        <a href="#" target="_blank" title="查看"></a>
        <a href="#" title="编辑"></a>
        <a href="#" title="复制"></a>
        <a href="#" title="回收站"></a>
        <a href="#" title="货品列表"></a>
    </td>
</tr>
</foreach>
....

```

10.9 商品的删除

1). 模版中进行相应的地址传参

goodslist.html

```

....
<foreach name='goods' item='g'>
<tr>
    <td><input type="checkbox" name="checkboxes[]" value="32" />{$g.goods_id}</td>

    <td class="first-cell" style=""><span>{$g.goods_name}</span></td>
    <td><span>{$g.goods_sn}</span></td>
    <td align="right"><span>{$g.shop_price}</span></td>
    <td align="center"></td>
    <td align="center"></td>
    <td align="center"></td>
    <td align="center"></td>

    <td align="center"><span>100</span></td>
    <td align="right"><span>1</span></td>
    <td align="center">

```

```

<a href="#" target="_blank" title="查看"></a>
<a href="#" title="复制"></a>
<a href="{:U('Admin/Goods/del',array('goods_id'=>$g['goods_id']))}" title="回收站"></a>
</tr>
</foreach>
....

```

2).执行模块的删除并且调到列表页

```

....
public function del(){
$model = D('Goods');
//$rs = $model->delete($_GET['goods_id']);
$rs = $model->delete(I('get.goods_id'));
if ($rs) {
$this->redirect('Admin/Goods/goodslist');
}else{
echo "删除失败";
}
}
....

```

10.10 无限极分类[可直接复制]

学习方法:重点在于表的设计,表设计好了,无限分类就不难.

关键在于:parent_id

画图详解

1).先把所有的栏目都查出来,利用自动执行

```

class CatModel extends Model{
protected $cats = array();
public function __construct(){
parent::__construct();
$this->cats = $this->select();
}
public function getTree(){

}
}

```

2)先不要思考无限分类,先考虑如何把parent_id=0的顶层分类找出来

循环判断谁的parent_id=0

```

class CatModel extends Model{
protected $cats = array();
public function __construct(){
parent::__construct();
$this->cats = $this->select();
}
public function getTree($parent_id=0){
$tree = array();

foreach($this->cats as $c){
if($c['parent_id'] == $parent_id){
$tree[] = $c;
}
}
}
}

```


3)找出来顶层栏目,肯定它下面还有下层栏目,第一次循环如果出现0号的parent_id顺手再跑到下层栏目下面找找还有没有别的栏目

```
class CatModel extends Model{
    protected $cats = array();
    public function __construct(){
        parent::__construct();
        $this->cats = $this->select();
    }
    public function getTree($parent_id=0){
        $tree = array();

        foreach($this->cats as $c){
            if($c['parent_id'] == $parent_id){
                $tree[] = $c****;
            }
        }
    }
}
```

4).再查出下层的目录拼接到一起去,再调用自己,把查出来的cat_id放进去再查

```
public function getTree($parent_id=0){
    $tree = array();
    foreach($this->cats as $c){
        if ($c['parent_id'] == $parent_id) {
            $tree[] = $c;

            //再查出下层的目录拼接到一起去
            $tree = array_merge($tree,$this->getTree($c['cat_id']));
        }
    }
    return $tree;
}
```

5).在catController中catelist调用getTree方法

```
....
public function catelist(){
    $catModel = D('Cat');
    $catlist = $catModel->getTree();

    $this->assign('list',$catlist);
    $this->display();
}
....
```

6).但还是没有出现分层的显示,看不出来,稍微改动即可

```
public function getTree($parent_id=0,$lev=0){
    $tree = array();
    foreach($this->cats as $c){
        if ($c['parent_id'] == $parent_id) {
            $c['lev'] = $lev;
            $tree[] = $c;

            //再查出下层的目录拼接到一起去
            $tree = array_merge($tree,$this->getTree($c['cat_id'] , $lev+1));
        }
    }
    return $tree;
}
```

7).在cateadd中调用方法并在模板中修改,里面有一个magegin-left:0em,只需要把0替换成我们需要的\$lev的值就可以了

```
public function catelist(){
    $catModel = D('Cat');
    $catlist = $catModel->getTree();
    //print_r($catlist);
    $this->assign('list',$catlist);
    $this->display();
}
```

```
<foreach name='list' item='ls'>
    <tr align="center" class="0" id="0_1" id = 'tr_1'>
    <td align="left" class="first-cell" style = 'padding-left="0"'>
        
        <span><a href="#" >{$ls.cat_name}</a></span>
    </td>
    <td width="10%">0</td>
```

第11章 TP功能库[full版本]

11.1 上传功能

类: \Think\Upload

- 注意: 要指定上传文件的"根路径",即rootPath;
- 返回值: 失败为false,成功为2维数组

例:

```
array('goods_img'=>array('savepath'=>'xxx','savename'=>'yyy'))
```

案例:

```
$upload = new \Think\Upload();
$upload->exts = array('jpg','jpeg','png');

//在public下创建image/Y-m-d这样格式的子目录
$upload->savePath = './Public/up/';
$info = $upload->upload();
```

实战:

先查看手册对应着看相关的功能,这个时候你要想到我们之前写的上传类有一个函数叫什么?

- 1).对着手册进行实战,先看对应的模版中是否有上传相关的信息,回忆,上传都需要哪些对应的HTML代码
- 2).进行C层相关的操作Goods/add

```
public function Goodsadd(){
    if(!IS_POST){
        $this->display();
    }else{
        $goods = D('Goods');
        $upload = new \Think\Upload();// 实例化上传类
        //$upload->maxSize = 3145728;// 设置附件上传大小
        $upload->exts = array('jpg', 'gif', 'png', 'jpeg');// 设置附件上传类型
        $upload->rootPath = './Public/up/'; // 设置附件上传根目录
        $info = $upload->upload();
        if (!$info) {
            var_dump($upload->getError());
        }else{
            echo "<pre>";
            var_dump($info);
        }
        exit();
        if(!$goods->create()){
```

```

    echo $goods->getError();
    exit();
}else{
    $goods->add();
}
}
}
}

```

- 2)发现报错了,上传根目录不存在! 请尝试手动创建:./Public/up根目录不存在,手动去创建
 3).拼写数据库地址.这里面要注意,要用到打印出来的数据,根据具体的参数进行地址添加

```

public function Goodsadd(){
    if(!IS_POST){
        $this->display();
    }else{
        $goods = D('Goods');
        $upload = new \Think\Upload();// 实例化上传类
        // $upload->maxSize = 3145728;// 设置附件上传大小
        $upload->exts = array('jpg', 'gif', 'png', 'jpeg');// 设置附件上传类型
        $upload->rootPath = './Public/up/'; // 设置附件上传根目录
        $info = $upload->upload();
        if (!$info) {
            var_dump($upload->getError());
        }else{
            //echo "<pre>";
            //var_dump($info);
            $_POST['goods_img'] = '/Public/Up/'.$info['goods_img']['savepath'].$info['goods_img']['savename'];
        }
        if(!$goods->create()){
            echo $goods->getError();
            exit();
        }else{
            $goods->add();
        }
    }
}
}

```

11.2 图片缩放

类:\Think\Image

调用流程:

```

// 1: 实例化
$img = new \Think\Image();

// 2: 打开图片
$img->open('原始图片的绝对路径');

// 3: 缩放保存
$img->thumb(宽,高,缩放选项)->save('保存路径');

```

手册 专题\图片处理 找缩放 操作并存放在数据库中

```

$_POST['goods_img'] = '/Public/up/'.$info['goods_img']['savepath'].$info['goods_img']['savename'];
$path = './Public/up/'.$info['goods_img']['savepath'].$info['goods_img']['savename'];

$img = new \Think\Image();
$img->open($path);
$thumb_path = '/public/up/thumb/'.$info['goods_img']['savename'];
$thumb_path1 = './public/up/thumb/'.$info['goods_img']['savename'];
$img->thumb(208,208)->save($thumb_path1);
$_POST['thumb_img'] = $thumb_path;

```

11.3 分页功能

类:\Think\Page

分页使用的正式limit

调用流程:

```
// 1:实例化,2个参数为总条目及每页显示数量
$pager = new \Org\MyPage($cnt , 5);
$page = $pager->show(); // $page为分页代码,显示到模版几个
```

实战:

```
public function goodsList(){
    $goodsmodel = D('Goods');
    $count = $goodsmodel->count();// 查询满足要求的总记录数
    $Page = new \Think\Page($count,10);// 实例化分页类 传入总记录数和每页显示的记录数(25)
    $show = $Page->show();// 分页显示输出
    // 进行分页数据查询 注意limit方法的参数要使用Page类的属性

    $goodsmodel->field('goods_id,goods_name,goods_sn,shop_price,is_on_sale,is_best,is_new,is_hot,goods_number');
    $goods = $goodsmodel->order('goods_id desc')->limit($Page->firstRow.','.$Page->listRows)->select();
    $this->assign('goods',$goods);
    $this->assign('page',$show);// 赋值分页输出
    //var_dump($goods);
    $this->display();
}
```

模版显示:

```
{ $page }
```

11.4 验证码

564

54

类:\Think\Verify

```
public function yzm(){
    $Verify = new \Think\Verify(); //实例化
    $Verify->imageW = 120; //设置宽
    $Verify->imageH = 40; //设置高
    $Verify->fontSize = 20; //设置字体大小
    $Verify->length = 4; //设置验证码字数
    $Verify->userNoise = false; //干扰线
    $Verify->entry(); //验证码输出
}
```

调用如下:

```
<span>验证码 </span><input type="text" name="yzm" value=""/>
```

验证办法:

另外写一个方法,因为验证码里面有name值,可以传过来

```
<form action="{:U('Home/User/checkyzm')}" method="post" name="loginForm" id="loginForm">
```

方法里面:

```
public function chectyzm(){
    $Verify = new \Think\Verify();
    if($Verify->check(I('post.yzm'))){
```

```
echo '验证码正确';
}else{
echo '验证码错误';
}
}
```

第12章 完成前台页面

12.1 查询热门商品

1).先观察热门商品栏目对应的字段

开始查询

\\Home\\index\\index

```
namespace Home\\Controller;
use Think\\Controller;
class IndexController extends Controller {
    public function index(){
        $hot_goods = D('Goods')->field('goods_id,goods_name,shop_price,goods_img,market_price')
            ->where('is_hot=1')->order('goods_id desc')->limit(4)->select();
        $this->assign('hots',$hot_goods);
        //var_dump($hot_goods);
        $this->display();
    }
}
```

2).在对应的模版中修改对应的变量

```
....
<foreach name="hots" item="hot">
    <div class="goodsbox1">
        <div class="imgbox1"><a href="./goods.php-id=13.htm"></a>
        <a href="./goods.php-id=13.htm" title="">{ $hot.goods_name}</a><br>
            <font class="goodspice market">{ $hot.market_price}</font>
            <b class="f1">{ $hot.shop_price}</b><br>
        </div>
    </foreach>
....
```

12.2 查询栏目下的商品

1).对应的cat模块之前已经让大家写好,但是它是从首页过来的,需要点过来.

之前写过一个无限分类getTree()遍历到前台

```
namespace Home\\Controller;
use Think\\Controller;
class IndexController extends Controller {
    public function index(){
        // 查询首页所有的栏目
        $tree = D('Admin/Cat')->getTree();
        $this->assign('tree',$tree);
        // 查询热门
        $hot_goods = D('Goods')->field('goods_id,goods_name,shop_price,goods_img,market_price')
            ->where('is_best=1')->order('goods_id desc')->limit(4)->select();
        $this->assign('hots',$hot_goods);
        //var_dump($hot_goods);
        $this->display();
    }
}
```

2).模版修改

```

....
<div class="categoryCon clearfix">
  <foreach name="tree" item="tr1">
    <if condition="$tr1[parent_id] eq 0">
      <dl>
        <dt><a href="{:U('Home/Cat/cat',array('cat_id'=>$tr1['cat_id']))}">{$tr1['cat_name']}</a></dt>
        <dd class="clearfix">
          <foreach name='tree' item='tr2'>
            <if condition="$tr2['parent_id'] eq $tr1['cat_id']">
              <p class=""><a href="{:U('Home/Cat/cat',array('cat_id'=>$tr2['cat_id']))}" title="CDMA手机" class="txttdot">{$tr2['cat_name']}</a></p>
            </if>
          </foreach>
        </dd>
      </dl>
    </if>
  </foreach>
</div>
....

```

3).值传过来以后,我需要接受一下cat_id传过来的值然后做相关操作

```

namespace Home\Controller;
use Think\Controller;
class CatController extends Controller{
    public function cat(){
        $cat_id = I("get.cat_id");
        $cats= D('goods')->field('goods_id,goods_name,shop_price,goods_img,market_price')
            ->where("cat_id=$cat_id")->select();
        //var_dump($cats);
        $this->assign('cat_goods',$cats);
        $this->display();
    }
}

```

4).模板中继续之前的修改操作

```

<div class="clearfix modContent">
  <form name="compareForm" action="" method="post">
    <div class="clearfix grid">
      <foreach name="cat_goods" item="cats">
        <div class="goodsbox1" style="margin: 5px 9px 8px 8px;*margin:5px 6px 10px 14px;">
          <div class="imgbox1"><a href=""></a></div>
          <a href="" title="">{$cats['goods_name']}</a><br>
          <font class="market">{$cats['market_price']}</font>
          <b class="f1">{$cats['shop_price']}</b><br>
        </div>
      </foreach>
    </div>
  </form>
</div>

```

12.3 商品页面

1).修改准确的每个商品的页面

```

....
<a href="{:U('Home/goods/goods',array('goods_id'=>$cats['goods_id']))}"></a>
....

```

2).通过ID进行C操作

```

....

```

```
public function goods(){
    $this->assign('goods',D('goods')->find(I('get.goods_id')));
    $this->display();
}
....
```

3).进行模版操作(学员完成)

12.4 面包屑

首先要明白什么是面包屑,导航,主要原理是:获取上一层,上上一层的名称,注意观察表和实际内容显示的信息有何异同

1).goods里面新建mbx方法

```
public function mbx($cat_id){
    $row = D('cat')->find($cat_id);
    $tree[] = $row;

    while($row['parent_id']>0){
        $row = D('cat')->find($row['parent_id']);
        $tree[] = $row;
    }
    return array_reverse($tree);
}
```

2).修改对应的显示到模版中的方法,这里用的是goods方法显示

```
public function goods(){
    $goods_info = D('goods')->find(I('get.goods_id'));
    //var_dump($this->mbx($goods_info['cat_id']));
    $this->assign('mbx',$this->mbx($goods_info['cat_id']));
    //exit();
    $this->assign('goods',$goods_info);
    $this->display();
}
```

3).显示到模版中

goods.html

```
<div class="ur_here blank">
当前位置:
<a href="">首页</a>
<foreach name='mbx' item='x'>
<a href="">{$x['cat_name']}</a>
<code>&gt;</code>
</foreach>
{$goods['goods_name']}
</div>
```

13 cookie/session处理

13.1 会员注册验证

1).先有表

```
create table user (
user_id int unsigned auto_increment primary key,
username char(30) not null default '',
email char(30) not null default '',
password char(32) not null default '',
salt char(8) not null default '',
key username(username),
```

```
key email(email)
) engine myisam charset utf8;
```

2).修改模版和提交地址

先去模版跟数据库字段名字是否相同,然后提交到user\reg进行验证

```
....
<form action="{:U('Home/User/reg')}}" method="post" name="formUser" >
....
```

3).提交后做个验证

新建UserModel.class.php

```
namespace Home\Model;
use Think\Model;

class UserModel extends Model{
    public $_Validate = array(
        array('username','3,10','用户名长度必须3到10位',1,'length',3),
        array('email','email','邮箱不合法',1,'regex',3),
        array('password','6,16','密码长度必须6到16位',1,'length',3),
        array('re_password','password','两次密码不一致',1,'confirm',3)
    )
}
```

4).进行验证的C层判断

```
public function reg(){
    if(!IS_POST){
        $this->display();
    }else{
        $userModel = D("User");
        if(!$userModel->create()){
            echo $userModel->getError();
        }
    }
}
```

13.2 注册密码加盐

1).用随机数生成盐

```
public function yan(){
    return mt_rand(1000,9999);
}
```

2).需要在成功以后调用一下,拿出盐,提供给post传递过来的密码使用

```
public function reg(){
    if(!IS_POST){
        $this->display();
    }else{
        $userModel = D("User");
        if(!$userModel->create()){
            echo $userModel->getError();
        }else{
            $yan = $this->yan();
            //把post传递过来password的明文密码加盐传递给password
            $userModel->password = md5($userModel->password.$yan);
            //把生成的盐也加入到对应的表名中
            $userModel->salt = $yan;
            //凑齐了所有的字段,假如到数据库中
```



```

        if($userModel->add()){
            $this->redirect('Home/User/login');
        }
    }
}
}
}

```

13.3 用户登录和退出

1).上一节课我们提交到login里面去了,因为里面做了post判断,所以模版中也需要修改,当点击提交的时候,把信息提交到login中进行登录相关操作

2).当提交过来以后,先验证码验证,我们有一个方法叫check_yzm()方法

check_yzm修改

```

public function check_yzm($yzm){
    $Verify = new \Think\Verify();
    if($Verify->check($yzm)){
        return true;
    }else{
        echo "验证码错误";
        exit();
    }
}
}
}

```

```

public function login(){
    if (!IS_POST){
        $this->display();
    }else{
        //var_dump($_POST);
        //验证码成功以后再进行密码+盐验证
        if($this->check_yzm(I('post.yzm'))){

        }

    }
}
}

```

3).验证码通过以后再进行加盐密码的验证

```

....
if($this->check_yzm(I('post.yzm'))){
    //进行用户名相关查询
    $user = D('User')->where('username="'.I('post.username').'"')->find();
    //查出来的数据跟用户传过来同样加盐以后md5的码进行对比
    if(md5(I('post.password').$user['salt']) === $user['password']){

    }else{
        echo "密码或者用户名不对";
    }
}
}
....

```

4).写入cookie

```

if($this->check_yzm(I('post.yzm'))){
    //进行用户名相关查询
    $user = D('User')->where('username="'.I('post.username').'"')->find();
    //查出来的数据跟用户传过来同样加盐以后md5的码进行对比
    if(md5(I('post.password').$user['salt'])==$user['password']){
        //登录成功以后写入cookie
        cookie('username',$user['username']);
        $this->redirect('/');
    }else{

```

```

    echo "密码或者用户名不对";
}

}

```

5).退出功能晚上
修改模版

```

<font id="ECS_MEMBERZONE">
    您好，欢迎光临本店！
    <if condition="$Think.cookie.username eq null ">
    <a href="{:U('Home/User/login')}" style="color:#50884b">登录</a> |
    <a href="{:U('Home/User/reg')}" style="color:#50884b">免费注册</a>
    <else />
    {$Think.cookie.username} |
    <a href="{:U('Home/User/logout')}" style="color:#50884b">退出</a>
    </if>
</font>

```

6).写退出方法

```

public function logout(){
    cookie('username',null);
    $this->redirect('/');
}
}

```

因为所有的页面都需要cookie 的信息,需要引入相应的模版

13.4 cookie加盐

我们同样需要在cookie里面加盐

1).我们需要在config里面写一段盐

```

return array(
    // '配置项'=>'配置值'
    'DB_TYPE'          => 'mysql',      // 数据库类型
    'DB_HOST'          => '127.0.0.1',  // 服务器地址
    'DB_NAME'          => 'shop',       // 数据库名
    'DB_USER'          => 'root',       // 用户名
    'DB_PWD'           => 'root',       // 密码
    'salt'             => 'dffdgfgf',   // cookie加盐
);

```

2),因为之前有一个cookie,我需要再写一个加了username并且加了盐的cookie

```

....
// 查出来的数据跟用户传过来同样加盐以后md5的码进行对比
if(md5(I('post.password')).$user['salt'])==$user['password']){
    // 登录成功以后写入cookie
    cookie('username',$user['username']);
    cookie('coode',md5($user['username'].C('salt')));
    $this->redirect('/');
}else{
    echo "密码或者用户名不对";
}
....

```

3).因为我们在模版里面只判断了一个username,并且判断的时候用的null,我现在换成true或者false,或者0,1,如何判断这个盐是登录的时候加上的而不是自己修改的?

写一个第三方的函数,登录完了以后再模板中调用这个第三方的函数,这个函数的主要功能是验证你cookie加了盐的那个乱码跟用户登录时的那个乱码是否相等

Home\Common\function.php ---- 系统自己会识别function.php名字

```
function cookie_check(){
    if(md5(cookie('username').C('salt')) === cookie('coode')){
        return 1;
    }else{
        return 0;
    }
}
```

4).当你需要登录时,我就写入一个加了盐的cookie,这个时候,在模版中就需要调用一下这个函数,这个函数会去判断,判断完了以后给你一个0或者1,1表示cookie一样,0表示你cookie是乱改的
再进行登录

```
<font id="ECS_MEMBERZONE">
    您好, 欢迎光临本店!
    <if condition="cookie_check() eq 1 ">
    {$Think.cookie.username} |
    <a href="{:U('Home/User/logout')}"" style="color:#50884b">退出</a>

    <else />
    <a href="{:U('Home/User/login')}"" style="color:#50884b">登录</a> |
    <a href="{:U('Home/User/reg')}"" style="color:#50884b">免费注册</a>

    </if>
</font>
```

5).别忘了退出的时候也需要删除cookie

```
public function logout(){
    cookie('username',null);
    cookie('coode',null);
    $this->redirect('/');
}
```

13.5 session浏览历史

首先要明白,我们为什么要做历史相关的操作,步骤是什么?要出什么效果,以后遇到新问题,不要怕,有问题就想办法出去问题,一步一步来,不要着急,先去培养解决问题的能力,而不是问人的能力

1).需要在goods()查询出来的信息做判断,因为当你查询出来有信息的时候才进行写入session,否则不写入

```
public function goods(){
    $goods_info = D('goods')->find(I('get.goods_id'));
    if($goods_info){
        $this->history($goods_info);
        //var_dump(session('history'));
    }

    //var_dump($this->mbx($goods_info['cat_id']));
    $this->assign('mbx',$this->mbx($goods_info['cat_id']));
    //exit();
    $this->assign('goods',$goods_info);
    $this->display();
}
```

2).写对应的方法

```
public function history($info){

    //2.需要判断session里面有没有历史,没有给空,有的话把session赋给$row
    $row = session('?history')?session('history'):array();

    //1.把你查过的数据都放在一个数组中,然后把查出来对应的商品的id新建一个数组,形成一个二维数组
```

```

$g = array();
$g['goods_name'] = $info['goods_name'];
$g['shop_price'] = $info['shop_price'];
$g['goods_id'] = $info['goods_id'];
$row[$info['goods_id']] = $g;

//假如大于5条,我就删除第一条
if(count($row) > 5){
    $key = key($row);
    unset($row[$key]);
}

session('history',$row);
}

```

3)因为我们的浏览历史写到cat页面了里面了,需要在cat对应的方法里面显示出历史浏览,顺序不太对,需要用到翻转函数

```

public function cat(){
    $cat_id = I("get.cat_id");
    $this->assign('his',array_reverse(session('history')));
    $cats= D('goods')->field('goods_id,goods_name,shop_price,goods_img,market_price')->where("cat_id=$cat_id")->select();
    //var_dump($cats);
    $this->assign('cat_goods',$cats);
    $this->display();
}

```

4),写到对应的模版中

cat.html

```

<h1 class="mod2tit" style="background:url(__PUBLIC__/Home/images/sdgg.gif) repeat-x; height:27px; color:#000">浏览历史</h1>
<div class="mod1 mod2 blank" id="topbox">
<span class="lb"></span><span class="rb"></span>
<ul id="top10">
    <foreach name='his' item='h'>
        <li>
            <div class="first">
                <div class="f1">
                    <font style="color:#F00; font-weight:bold">{$h['goods_id']}</font> <a href="" title="">{$h['goods_name']}</a>
                </div>
                <div class="fr"><b class="f1">¥{$h['shop_price']}元</b></div>
            </div>
        </li>
    </foreach>
</ul>
</div>

```

14章 第3方库的导入

对于controller,Model之外,不可避免的还会有很多其他的类.
如邮件发送,购物车.

说明: APP代表项目的路径

APP/Module下面, 如/www/shop/Home下

放置自定义的目录,如Tool

Tool下的文件必须叫XxxTool.class.php

类名称叫: class XxxTool {}

然后通过 new \Home\XxxTool();

例如:

Home\Tool

```
namespace Home\Tool;
class CeTool{
    public function ha(){
        return 123;
    }
}
```

随便一个C层去测试,这里用index

```
public function xx(){
    //new \Home\Tool\CeTool();
}
```

也可以用

```
use Home\Tool\CeTool;
....
$ceshi = new CeTool();
echo $ceshi->ha();
....
```

第15章 购物下单

15.1 订单号

```
#订单信息
create table ordinfo (
    ordinfo_id int unsigned primary key auto_increment,
    ord_sn char(12) not null default '',
    user_id int not null default 0,
    xm char(10) not null default '',
    mobile char(11),
    address varchar(30),
    paytype tinyint not null default 1, #1在线,2是否到付
    paystatus tinyint not null default 0, #0未支付,1已付款
    money decimal(9,2),
    note varchar(20) not null default '',
    ordtime int unsigned not null default 0,
    key user_id(user_id)
)engine myisam charset utf8;
```

orinfo ---- 订单主键

ord_sn ---- 订单号

user_id ---- 用户ID

xm ---- 下订单的人名

mobile ---- 手机

address ---- 地址

pagtype ---- 支付类型

paystatus ---- 支付状态

money ---- 总金额

note ---- 备注

ordtime ---- 订单时间

```
#订单对应的商品
create table ordgoods (
    ordgoods_id int primary key auto_increment,
    ordinfo_id int not null default 0,
    goods_id int not null default 0,
    goods_name varchar(20),
    shop_price decimal(7,2),
    goods_num smallint unsigned not null default 0,
```

```
key ordinfo_id(ordinfo_id)
)engine myisam charset utf8;
```

ordgoods_id ---- 商品主键

ordinfo ---- 关联上一张表的订单主键

goods_id ---- 商品id

goods_name ---- 商品名字

shop_price ---- 商品价格

goods_num ---- 购买数量

15.2 购物车类

功能:

- 添加1个商品
- 减少1个商品的数量
- 删除1个商品
- 列出购物车的商品种类
- 计算购物车的商品数量
- 计算购物车中的总金额
- 清空购物车

抽象类代码:

```
namespace Home\Tool;
abstract class ACartTool {
    /**
     * 向购物车中添加1个商品
     * @param $goods_id int 商品id
     * @param $goods_name String 商品名
     * @param $shop_price float 价格
     * @return boolean
     */
    abstract public function add($goods_id,$goods_name,$shop_price);

    /**
     * 减少购物车中1个商品数量,如果减少到0,则从购物车删除该商品
     * @param $goods_id int 商品id
     */
    abstract public function decr($goods_id);

    /**
     * 从购物车删除某商品
     * @param $goods_id 商品id
     */
    abstract public function del($goods_id);

    /**
     * 列出购物车所有的商品
     * @return Array
     */
    abstract public function items();

    /**
     * 返回购物车有几种商品
     * @return int
     */
    abstract public function calcType();

    /**
     * 返回购物车中商品的个数
     * @return int
     */
    abstract public function calcCnt();
```

```

/**
 * 返回购物车中商品的总价格
 * @return float
 */
abstract public function calcMoney();

/**
 * 清空购物车
 * @return void
 */
abstract public function clear();
}

class CarTool extends ACartTool{
    public $item = array();
    /**
     * 向购物车中添加1个商品
     * @param $goods_id int 商品id
     * @param $goods_name String 商品名
     * @param $shop_price float 价格
     * @return boolean
     */
    public function add($goods_id,$goods_name,$shop_price){
        if(isset($this->item[$goods_id])){
            $this->item[$goods_id]['num']+=1;
        }else{
            $goods = array('goods_name'=>$goods_name,'shop_price'=>$shop_price,'num'=>1);
            $this->item[$goods_id] = $goods;
        }
    }

    /**
     * 减少购物车中1个商品数量,如果减少到0,则从购物车删除该商品
     * @param $goods_id int 商品id
     */
    public function decr($goods_id){
        //先判断购物车中有没有商品,有的话减1
        if(isset($this->item[$goods_id])){
            $this->item[$goods_id]['num'] -=1;
        }
        //再判断如果到0了,不能再减了,需要删除,调用的是下面del
        if($this->item[$goods_id]['num'] <= 0){
            $this->del($goods_id);
        }
    }

    /**
     * 从购物车删除某商品
     * @param $goods_id 商品id
     */
    public function del($goods_id){
        unset($this->item[$goods_id]);
    }

    /**
     * 列出购物车所有的商品
     * @return Array
     */
    public function items(){
        return $this->item;
    }

    /**
     * 返回购物车有几种商品
     * @return int
     */

```

```

public function calcType(){
    //因为有几个goods_id就有几个商品
    return count($this->item);
}

/**
 * 返回购物车中商品的个数
 * @return int
 */
public function calcCnt(){
    $cnt = 0;
    //拿出多少种商品,每一种商品有多少件
    foreach($this->item as $v){
        $cnt += $v['num'];
    }
    return $cnt;
}

/**
 * 返回购物车中商品的总价格
 * @return float
 */
public function calcMoney(){
    $money = 0;
    //拿出多少种商品,每一种商品有多少件*每件商品的价格
    foreach($this->item as $v){
        $money += $v['num']*$v['shop_price'];
    }
    return $money;
}

/**
 * 清空购物车
 * @return void
 */
public function clear(){
    $this->item = array();
}
}

```

1).写完类以后,需要到把你要买的每一件商品放在购物车中
Controller\FlowController.class.php

```

namespace Home\Controller;
use Think\Controller;

class FlowController extends Controller{
    public function add(){

    }
}

```

2).修改goods.html模版,让每一个商品都可以调转到这个控制器里,进行相关的购物车操作.

```

....
<a href="{:U('Home/Flow/add',array('goods_id'=>$goods['goods_id']))}"></a>
....

```

3).对应的写方法

```

namespace Home\Controller;
use Think\Controller;

class FlowController extends Controller{

```



```

public function add(){
    $goodsinfo = D('goods')->find(I('get.goods_id'));
    //var_dump($goodsinfo);
    $car = new \Home\Tool\CarTool();
    //加多少都白搭,因为脚本运行到初次结束了,下次还得继续从头执行,而且可以多次new
    $car->add($goodsinfo['goods_id'],$goodsinfo['goods_name'],$goodsinfo['shop_price']);
    var_dump($car->items());
}
}

```

4).修改购物车,改成单例模式,你new多了就出多个购物车了.

```

class CarTool extends ACartTool{
    //2.放属性的静态属性
    public static $ins = null;
    public $item = array();

    //4.自动运行,而且不可以继承重写,你new就自动执行__construct方法,就报错了,而final不可以继承
    final protected function __construct(){
        //暂时不用写东西
    }

    //1.先改成静态的,不让外部new然后放在静态属性里面,但可以重写方法
    public static function getIns(){
        //3.进行判断
        if(self::$ins == null){
            self::$ins = new self();
        }
        return self::$ins;
    }
    .....
}

```

5).不可以new了,对应的方法也应该修改

```

public function add(){
    $goodsinfo = D('goods')->find(I('get.goods_id'));
    //var_dump($goodsinfo);
    $car = \Home\Tool\CarTool::getIns();
    $car->add($goodsinfo['goods_id'],$goodsinfo['goods_name'],$goodsinfo['shop_price']);
    var_dump($car->items());
}
}

```

6).脚本结束以后都没了,我需要用析构方法当类结束时帮我把信息存下来,用的是session

```

class CarTool extends ACartTool{
    //2.放属性的静态属性
    public static $ins = null;
    public $item = array();

    //4.自动运行,而且不可以继承重写,你new就自动执行__construct方法,就报错了,而final不可以继承
    final protected function __construct(){
        //6.需要判断session有没有值,如果之前结束的时候有值,说明你之前买过东西,我就把你session的东西取出来放在item里面
        if(session('?cart')){
            $this->item = session('cart');
        }
    }

    //1.先改成静态的,不让外部new然后放在静态属性里面,但可以重写方法
    public static function getIns(){
        //3.进行判断
        if(self::$ins == null){
            self::$ins = new self();
        }
        return self::$ins;
    }
    .....
}

```

```
public function __destruct(){
    //5.你虽然存了,但还是没有取出来啊,你下次再添加的时候还有之前的信息

    session('cart',$this->item);
}
```

15.3 购物车展示

1).对应的方法是c层Goods里面checkout方法

```
public function checkout(){
    $this->assign('che',session('cart'));
    $this->display();
}
```

2).修改对应的模版

```
<foreach name='che' item='c'>
    <tr>
        <td bgcolor="#ffffff">
            <a target="_blank">{$c['goods_name']}</a>
        </td>
        <td bgcolor="#ffffff">颜色:黑色 <br></td>
        <td align="right" bgcolor="#ffffff">¥3612.00元</td>
        <td class="price" align="right" bgcolor="#ffffff">¥{$shop_price}元</td>
        <td align="right" bgcolor="#ffffff">{$c['num']}</td>
        <td class="price" align="right" bgcolor="#ffffff">¥{$c['num']*$c['shop_price']}元</td>
    </tr>
</foreach>
```

剩下的学员自己完成.注意:是两张表,分别进行添加.

```
public function done() {
    $oi = M('ordinfo');
    $og = M('ordgoods');
    $cart = \Home\Tool\CartTool::getIns();

    //写入到ordinfo
    $_POST['ord_sn'] = date('Ymd') . rand(1000,9999);
    $_POST['user_id'] = $_COOKIE['user_id']?$_COOKIE['user_id']:0;
    $_POST['money'] = $cart->calcMoney();
    $_POST['ordtime'] = time();
    $ordinfo_id = NULL;
    $oi->create() && ($ordinfo_id=$oi->add());

    if(!$ordinfo_id) {
        $this->show('下单失败' , 'utf-8');
        exit;
    }

    //写入到ordgood表
    $data = true;
    foreach($cart->items() as $k=>$v) {
        $row = array();
        $row['ordinfo_id'] = $ordinfo_id;
        $row['goods_id'] = $k;
        $row['goods_name'] = $v['goods_name'];
        $row['shop_price'] = $v['shop_price'];
        $row['num'] = $v['num'];
        $data[] = $row;
    }

    if(!$og->addAll($data)) {
```

```

$oi->delete($ordinfo_id);
$log->where(array('ordinfo_id'=>$ordinfo_id))->delete();
$this->show('下单失败~~' , 'utf-8');
exit;
}

//清空购物车
$cart->clear();
$this->display();
}

```

第16章 在线支付

16.1 提交支付

```

<form action="网银/块钱/支付宝" method="post">
订 单 号: 123654
金    额: 25.60
数字签名:9F316Hi340GLBF237ERT
立即支付
</form>

```

```

/**
 * @file Home/Pay/CBPay.class.php
 */
/**
 * 网银在线<chinabank.com.cn> 支付类
 */

```

```

class CBPay {
protected $v_mid;
protected $paykey;
protected $v_md5info;
protected $v_url;
public $v_moneytype = 'CNY';
public $v_oid;
public $v_amount;
public function __construct() {
//商户号,支付密钥,回调地址写到配置文件中
$this->v_mid = C('v_mid');
$this->paykey = C('paykey');
$this->v_mid = C('v_url');
}

public function form() {
$this->crypt(); //生成数字签名

$str = '<form method=post action="https://pay3.chinabank.com.cn/PayGate">
<input type=hidden name=v_mid value="%s">
<input type=hidden name=v_oid value="%s">
<input type=hidden name=v_amount value="%s">
<input type=hidden name=v_moneytype value="%s">
<input type=hidden name=v_url value="%s">
<input type=hidden name=v_md5info value="%s">
<input type="submit" value="支付" />
</form>';

return sprintf($str , $this->v_mid , $this->v_oid , $this->v_amo
unt , $this->v_moneytype , $this->v_url , $this->v_md5info);
}

public function crypt() {

```

```
$str = $this->v_amount.$this->v_moneytype.$this->v_oid.$this->v_mid.$this->v_url.$this->paykey;
$this->v_md5info = strtoupper(md5($str));
}
}
```

16.2 数字签名

问题的关键: 支付平台如何确认本次支付提交的真实性,如金额,商家?

解决的办法: 数字签名

数字签名的原理:

1. 有1个秘钥,假设为"????", 具体内容商家知道,支付平台知道
2. 表单中数字签名 == md5(订单号+金额+商户号+????)
3. 支付平台计算 数字签名 == md5(订单号+金额+商户号+????)
4. 平台判断 计算得的数字签名 == 表单传来的数字签名

16.3 对账功能

在支付完成后,支付平台,会立即往回调地址POST数据

包含订单账号,商户号,金额,以及支付结果.

回调地址接收信息,并判断支付结果,

根据结果,修改订单的支付状态

第17章 关联模型

17.1 商品评论功能

```
create table comment (
comment_id int unsigned auto_increment primary key,
goods_id int unsigned not null default 0,
user_id int unsigned not null default 0,
email char(30) not null default '',
content varchar(140) not null default '',
pubtime int unsigned not null default 0
) engine myisam charset utf8;
```

comment_id ---- 评论ID

goods_id ---- 商品id

user_id ---- 用户id

email ---- 用户email

content ---- 评论内容

pubtime ---- 评论时间

****注意思考:****按照我们blog的评论做法,每一个文章下面可以有多条评论,叫一对多.这个我们是做过的,用的是两表联查,而TP中,使用的是关联模型,比较智能,tp自己可以判断你的哪张表有多个goods_id

17.2 继承关联模型

要使用TP的关联查询效果,需要继承关联模型

```
namespace Home\Model;
use Think\Model\RelationModel;
class GoodsModel extends RelationModel {
}
```

17.3 声明关联关系

一对一 (一个用户表和一个用户的详细信息)

一对多 (一个商品对应多个评论)

多对多 (多个标签对应多个视频)

利用Model的\$_link来说明关系.

以商品表goods为例,每条商品在comment表中,有多条评价.

```
protected $_link = array(
    'comment'=> self::HAS_MANY, //此处是个静态的本类,继承过来以后用self
);
```

在controller如下调用:

```
$goods = D('/Home/Goods');
$goods->find(23); //查出23号商品
$comms = $goods->relationGet('comment'); // 查出多条相关评论.
```

注意: 自动关联查询评价时,自动用goods表的主键来做连接.

上例中,goods表的主键是goods_id,则关联时产生的sql如下:

```
select * from comment where goods_id = 23;
```

如果评价内容比较多,建议直接用comment的Model操作,并指定查询字段.

一对一时,可以用关联模型简化操作.

实战操作:

1).GoodsModel.class.php里面新建

```
namespace Home\Model;
use Think\Model\RelationModel;

class GoodsModel extends RelationModel{
    protected $_link = array(
        'comment'=>self::HAS_MANY //这是一个常量,声明是一对多关系,并且Relation已经内部集成了Model功能了
    );
}
```

2).在对应的goods商品查询里面修改代码

```
public function goods(){

    $goods = D('Goods');//先用D方法连接数据库
    $goods_info = $goods->find(I('get.goods_id'));//进行查询goods表内容
    $goods_comment = $goods->relationGet('comment');//此时要使用relationGet方法,它会自动comment表,并把里面带goods_id的内容取出来
    //var_dump($goods_comment);
    //exit();

    if($goods_info){
        $this->history($goods_info);
        //var_dump(session('history'));
    }

    //var_dump($this->mbx($goods_info['cat_id']));
    $this->assign('mbx',$this->mbx($goods_info['cat_id']));
    //exit();
    $this->assign('goods_c',$goods_comment);
    $this->assign('goods',$goods_info);
    $this->display();
}
```

3).修改对应的模版

```
....
```

```

<div class="tips">
  <if condition='$goods_c eq null' >
    暂时还没有任何用户评论
  <else />
    <foreach name='goods_c' item='pl'>
      { $pl['content'] }<br>
    </foreach>
  </if>
</div>
....

```

4).把评论也顺手写了.跟之前同样的方法,注意,有个隐藏域,你得把哪个商品(goods_id)传过去.
模版修改

```

....
<form action="{:u('Home/Goods/addComment')}" method="post" name="commentForm" id="commentForm">
  <table border="0" cellspacing="5" cellpadding="0">
    <tbody><tr>
      <td align="right">用户名: </td>
      <td>匿名用户</td>
    </tr>
    <tr>
      <td align="right">E-mail: </td>
      <td>
        <input type="text" name="email" id="email" maxlength="100" value="" class="InputBorder">
      </td>
    </tr>
    <tr>
      <td align="right" valign="top">评论内容: </td>
      <td>
        <textarea name="content" class="InputBorder" cols="80" rows="3"></textarea>
        <input type="hidden" name="goods_id" value="{ $goods['goods_id'] }">
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <input name="" type="submit" value="提交评论" class="bnt_number4 fr">
      </td>
    </tr>
  </tbody></table>
</form>
....

```

写对应的方法 Goods里面

```

public function addComment(){
  if(D('comment')->add($_POST)){
    $this-> success('添加成功');
  }
}

```

第18章 编辑器

ckeditor(国外)
ueditor(百度)

第19章 URL美化

- 1.什么叫URL美化?
- 2.为什么要在URL上做美化工作

解释:很多做SEO的人认为这个URL不好看.而/模式呢,在linux下可能存在不兼容模式.

```
var_dump($_SERVER);
```

你在后面加/a/b/c 会发现有个path_info,TP就利用这个参数进行URL重写,再进行封装,因为我们的地址都是固定的index.php/模块/控制器/方法/id/3...你稍微写错了就报错找不到,TP正是利用这一点做到的.

另外,还有的SEO认为,你这个index.php也不美观,能不能把index.php去掉,网站域名后面直接跟模块/控制器/方法,更好看嘛

```
//默认false 表示URL区分大小写 true则表示不区分大小写
'URL_CASE_INSENSITIVE' => true,
// URL访问模式,可选参数0,1,2,3,代表4种模式:

// 0 (普通模式); 1 (PATHINFO模式); 2 (REWRITE 模式); 3 (兼容模式) 默认是1
'URL_MODEL' => 1,
// PATHINFO 模式下,各参数之间的分隔符
'URL_PATHINFO_DEPR' => '/',
```

0为普通模式

URL例: /index.php?m=Home&c=Index&a=goods&goods_id=34

3为兼容模式

URL例: /index.php?s=/Home/Index/goods/goods_id/34.html

1为pathinfo模式

URL例: /index.php/Home/Index/goods/goods_id/34.html

2为rewrite模式

URL例: /Home/Index/goods/goods_id/34.html

注意1: 此模式需要

webServer打开rewrite重写模块

以apache为例:

```
// http.conf 找到如下语句,去除行首#,并重启apache
#LoadModule rewrite_module modules/mod_rewrite.so
```

注意2: 并在网站根目录下写文件,名为'.htaccess'

```
<IfModule mod_rewrite.c>
RewriteEngine on RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php/$1 [QSA,PT,L]
</IfModule>
```

注意3: 上面URL重写用到的是正则表达式

第20章 缓存应用

20.1 什么是缓存cache?

短暂的存储.

比如,首页频率的查询"热门新闻",可以把查询结果缓存起来.

下次直接查询缓存,以避免大量的查询或计算.

20.2 缓存案例

Admin/Controller/CatController.class.php

method: add();

```
public function catelist(){
    $catModel = D('Cat');
```

```
$cat = S('catlist');
if($cat == false){
    echo '这是数据库出来的数据';
    $catlist = $catModel->getTree();
    S('catlist',$catlist,5);
}else{
    echo "这是缓存出来的数据";
    $catlist = $cat;
}
}
```

20.3 缓存有效期

既然有缓存,说明有一定的有效期.
有效期的长短根据具体业务逻辑而定.

如

S('cats', 'content', 300); //300秒

S('cats', 'content', 3600); //1小时

第20章 缓存应用

S('cats', NULL);

20.4 快速缓存

F(),和S一样,只是不用填有效期.
相当于手动删除之前,永久有效,

20.5 查询缓存

TP会封装很多方法去帮你拼装sql语句,最后组成mysql能识别的mysql语句,会加大查询的效率,有没有一种方法,可以帮我编译一次,组成mysql识别的语句,下次就不要拼装了.

比如: catModel

```
....
public function __construct(){
    parent::__construct();
    $this->cats = $this->cache(true)->select();
}
....
```

删除log日志,访问 cat/catelist查看日志变化