

# ThinkPHP学习

---

作者：西岭

Email: [xiling3105@163.com](mailto:xiling3105@163.com)

指导：燕十八

出自：布尔教育--高级PHP工程师培训

注：此文档不得转载，不得用于商业用途，不得修改传播；

如有其它用途或勘误，请以邮件方式联系作者；欢迎骚扰，违者必究!!!

## 框架的意义

---

- 减少重复劳动
- 便于团队配合
- 增强安全性
- 工作/面试都需要

## 为什么学习ThinkPHP

---

- 国内公司用的多
- 框架基本都是MVC架构,学一则通

## 框架的学习步骤

---

### 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/>

目前最新3.2.3

核心版core: 只有框架的MVC核心功能

完全版full: 有框架核心和实用类(上传,分页,验证码等)

## 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
```

注意: ThinkPHP中约定,目录必须以'/'结尾

## 1.3 生成项目

完成1.2的配置后,

在地址运行 index.php即可自动生成项目

看到如下界面,就配置成功了

```
: )
欢迎使用 ThinkPHP!
```

成功后,自动生成了如下目录

Common // 库,类等可以在这儿

Home // controll model view 都在这儿

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: 1个网站,常见前台, admin后台,二者功能,css,图片等,都差的很大.  
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

PHP namespace Home\Controller; 修改为 php 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

以典型的URL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

如果需要传递其他参数,可以如下写

index.php/模块/控制器/方法/id/3/price/9

### 2.2 控制器放在哪儿?

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: UserController.class.php

注意: 单词首字母大写[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class UserController
- 继承自谁? \Think\Controller
- 自身的空间叫什么? namespace和目录一致 示例: Home模块的下UserController

```
namespace Home\Controller;
use \Think\Controller;
class UserController extends Controller{
    public function reg() {
        echo '准备注册';
    }
}
```

## 第3章 引入模板 view

---

### 3.1 模板放在哪儿?

---

放在模块的view目录下

并且每个控制器的模板,要在与控制器同名的目录下.

以 index.php/Home/User/reg

则对应的模板在 /Home/view/User/reg.html

### 3.2 叫什么?

---

模板名和方法名是一致的

例: 地址栏: index.php/Home/User/reg

对应控制器: Home/UserController

对应的方法: reg

对应的模板: Home/View/User/reg.html

### 3.3 和控制器有什么对应关系?

---

如3.2

## 第4章 Model

---



## 4.1 放在哪儿?

项目/模块/Model目录下

以本教程为例,Home模块的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

如 GoodsModel.class.php 则对应操作 goods表

注意: 虽然Model类首字母大写的,但惯例,表名一律小写.

## 4.4 继承谁?

namespace Home\Model

use \Think\Model

```
class XxModel extends Model {  
}
```

## 4.5 在controller里如何实例化

`new \模块名\Model\ModelName();`

例:

```
new \Home\Model\UserModel();
```

也可以通过快捷函数D('模型名'),例:

```
D('User')
```

**注意:**

上述实验,都会失败,因为还没有配置数据库信息.

失败后,请参考第6章

## 区别

D(),M()函数都能得到Model的实例,

但D函数是优先得到子类的Model实例

而M函数是得到父类的Model实例,即\Think\Model类的实例

# 第5章 配置

## 5.1 配置文件在哪儿?

TP框架自带的配置文件:

```
/ThinkPHP/Conf/convention.php
```

项目的配置文件:

例项目在 /www/tpshop下,则项目配置为:

```
/www/tpshop/Common/Conf/config.php
```

模块的配置文件:

例Home模块,配置文件为:

```
/www/tpshop/Home/conf/config.php
```

## 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

在本例中,同样的配置选项,则:

模块的选项--覆盖-->项目同名配置--覆盖-->框架的同名配置

## 5.3 常用配置项

数据库选项

```
'DB_TYPE'           => 'mysql/oracle/sqlite', // 数据库类型,TP支持mysql,oracle,sqlserver,postgres及mongodb
'DB_HOST'           => '', // 服务器地址
'DB_NAME'           => '', // 数据库名
'DB_USER'           => '', // 用户名
'DB_PWD'            => '', // 密码
'DB_PORT'           => '', // 端口
```

```
'DB_PREFIX' => '', // 数据库表前缀
```

调试选项:

项目的入口页面,一般是index.php

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

```
// 记录日志
'LOG_RECORD' => true, // 默认不记录日志
'DB_SQL_LOG' => true, // SQL执行日志记录
```

## 第6章 Model操作

### 6.1 建立商品表和栏目表

```
CREATE TABLE goods (
  goods_id int(10) unsigned NOT NULL AUTO_INCREMENT,
  goods_sn char(15) NOT NULL DEFAULT '',
  cat_id smallint(6) NOT NULL DEFAULT 0,
  brand_id smallint(6) NOT NULL DEFAULT 0,
  goods_name varchar(60) NOT NULL DEFAULT '',
  shop_price decimal(9, 2) NOT NULL DEFAULT 0.00,
  market_price decimal(9, 2) NOT NULL DEFAULT 0.0,
  goods_number smallint(6) NOT NULL DEFAULT 11,
  click_count mediumint(9) NOT NULL DEFAULT 0,
  goods_weight decimal(6, 3) NOT NULL DEFAULT 0.000,
  goods_brief varchar(100) NOT NULL DEFAULT '',
  goods_desc text NOT NULL,
  thumb_img varchar(100) NOT NULL DEFAULT '',
  goods_img varchar(100) NOT NULL DEFAULT '',
  ori_img varchar(100) NOT NULL DEFAULT '',
  is_on_sale tinyint(4) NOT NULL DEFAULT 1,
  is_delete tinyint(4) NOT NULL DEFAULT 0,
  is_best tinyint(4) NOT NULL DEFAULT 0,
  is_new tinyint(4) NOT NULL DEFAULT 0,
  is_hot tinyint(4) NOT NULL DEFAULT 0,
  add_time int(10) unsigned NOT NULL DEFAULT 0,
  last_update int(10) unsigned NOT NULL DEFAULT 0,
  PRIMARY KEY ( goods_id ),
  UNIQUE KEY goods_sn ( goods_sn )
) ENGINE=MyISAM CHARSET=utf8;
```



```
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

1:面向过程风格

```
$user = array('name'=>'lilei','email'=>'lilei@q.com');  
$model->add($user);
```

2: 面向对象风格:

\$model->属性名 = 属性值

\$model->add();

例:

```
$m->name = 'lily';  
$m->email = 'lily@qq.com';  
var_dump($m->add())
```

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();

```
//select 查多行  
print_r($model->field('cat_id,catname')->where('cat_id>2')->order('cat_
```

```
id desc')->limit(0,3)->select());
```

## 6.4 更改数据

想改,先找那一行数据

```
$model->find(6);  
$model->catname = '测试修改-u1';  
var_dump($model->save());
```

## 6.5 删除数据

```
var_dump($model->delete(4)); // 按主键值来删除 where cat_id=4  
var_dump($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;
```

# 第7章 模板操作

## 7.1 assign赋值

```
$this->assign('title','布尔商城首页');  
$this->assign('bests',$bests);//$bests是二维数组
```

## 7.2 变量输出

在Html代码中输出标题如下:

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

注意:TP的模板标签和smarty稍有不同

TP在控制结构/循环/if/else时,用<>  
而最具体的输出变量 用{\$标签}

## 7.3 if/else标签

```
// 控制器代码
$this->assign('rand' , mt_rand(1,100));

// 模板标签
{$rand},<if condition="$rand egt 60">及格<else />不及格</if>
```

## 7.4 三元运算

```
// 控制器代码
$sel
{$goods['cat_id']==$v['cat_id']?"selected":""}
```

## 7.5 数组循环

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

## 7.6 import导入CSS,JS文件

import 标签,以APP/Public为起点,导入css,js

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

则导入

```
APP/Public/a/b/c.css
```

## 7.7 load导入css,js

例:

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

则导入 APP/Public/Js/Common.js

注: 7.6,7.7中的APP均指项目目录

## 7.8 模板包含

一般在模块下的view中,建立Public ,里面放置公共模板.  
如

```
/View/Home/Public/  
    header.html  
    footer.html
```

```
<include file="Public/模板名(不含后缀)" />  
例: 包含Home/View/Public/header.html:  
<include file="Public/header" />
```

## 7.9 不要让模板做太多工作

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



TP的Model父类,封装的功能比较多,增删改查操作都具备.  
一些表,比如留言表,comment

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

M('comment') 的作用就是得到父类Model的对象,  
且把Model的表名---指向-->comment

## 8.2 D函数

D函数也可以实例化Model,  
优先实例自定义的子类Model,  
如果子类Model不存在,则功能等同于M()的函数.

如果有自定义的类,也可以用D()来导入  
比如

```
/Home/  
    /Pay  
        AliPay.class.php  
        TenPay.class.php  
  
$ali = new \Home\Pay\AliPay();  
$ali = D("Ali","Pay");
```

## 8.3 D函数跨模块调用Model

如:在Admin模块下,调用 Home模块下的GoodsModel

```
D('Home/Goods')
```

## 8.4 U函数

U函数 按系统设定的URL规则,生成URL  
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/test',array('id'=>3));
U('Home/Index/test','id=5&page=3');
```

注意: 在模板中调用时,标签中加":"

例1:

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

例2:

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

例:

```
echo C('URL_PATHINFO_DEPR');
C('URL_PATHINFO_DEPR', '-');
echo C('URL_PATHINFO_DEPR');
```

## 8.6 I()函数

防止XSS攻击和sql注入,所做的过滤函数.

如:

```
index.php?title=<div>
```

```
I('get.title') --输出--> &lt;div&gt; // 防止XSS攻击
```

```
I('post.email') --输出-- 转义并过滤html标签后的email内容.
```

# 第9章 实战商城前台页面

## 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:/www/20150720/shop/web"
```

```
    ServerName shop.com
```

```
</VirtualHost>
```

## 9.2 引入模板

在Home/view下建立目录如下:

```
/Home/view
```

```
    /User
```

```
    /Index
```

```
    /Flow
```

分别把:

首页/栏目页/商品 存入 /Index

购物车系列模板 存入 /Flow

用户系列模板 存入 /User

## 9.3 写控制器

IndexController下在的方法

```
class IndexController extends Controller {
    public function index(){
        // 不加参数,默认找index.html做模板,即和自己的方法名一样
        $this->display();
    }

    public function cat() {
        // 也可以明确传递模板名称,但不要加".html"后缀
        $this->display('category');
    }

    public function goods() {
        $this->display();
    }
}
```

## 9.4 存放CSS/JS/图片资源

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:

```
/tpshop
/Public
```

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  
如:

```
/tpshop
/Public
/Home
```



## 9.5 修改模板中的资源路径

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

## 10.1 创建后台Index控制器

在Admin/view下建立目录如下:

```
/Admin/view  
  /Goods  
  /Cat
```

分别把:

商品管理模板 存入 /Goods

栏目管理模板 存入 /Cat

/Admin/Controller/IndexController

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

注意:把模板放在/Admin/View/Index目录下,自己整理好

## 10.2 栏目发布

```
public function add() {
    if(!IS_POST) {
        $this->display('cateadd');
        exit;
    }

    $cat = D('Home/Cat');
    //print_r($cat);

    if($cat->add($_POST)) {
        echo 'OK';
    } else {
        echo 'fail';
    }
}
```

## 10.3 栏目列表

```
public function lists() {
    $cat = D('Home/Cat');
    $cats = $cat->getTree(); ### 老师注意,一定要封装getTree()
    $this->assign('cats' , $cats);

    $this->display();
}
```

```
// /Admin/view/Cat/lists.html
<foreach name="cats" item="c">
</foreach>

// Home/Model/CatModel.class.php
public function getTree() {
    return $this->select();
}
```

## 10.3 栏目编辑和删除

```
public function edit() {
    $cat = D('Home/Cat');
    if(!IS_POST) {
        $row = $cat->find(I('get.cat_id'));
        $this->assign('row' , $row);
        $this->display();
    } else {
        echo $cat->where('cat_id='.I('post.cat_id'))->save($_POST);
    }
}

public function del() {
    $cat = D('Home/Cat');
    $cat->delete(I('get.cat_id'));
    $this->redirect('Admin/Cat/lists');
}
```

## 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]),  
    array(验证字段2,验证规则,错误提示,[验证条件,附加规则,验证时间]),  
    .....  
);
```

例:GoodsModel的验证规则

```
public $_validate = array(  
    array('goods_name','6,16','商品名称在6-16字符之间',1,'length',3),  
    array('is_best',array(0,1),'精品只能是0或1',0,'in',3),  
    array('cat_id','ckc','请先分类',1,'callback',3),  
    array('goods_sn','','goods_sn重复了',1,'unique',3),  
);  
  
protected function ckc() {  
    $cat = D('Home/Cat');  
    return $cat->find($_POST['cat_id']) ? true : false;  
}
```

注意:

function 则调用函数来验证.

callback 则调用Model自身的方法来验证.

**在一个项目,不可避免的会产生自定义函数,这些函数放在哪儿?**

**答:** TP3.2以后,要求命名为function.php (系统自带的叫functions.php)

放在 /APP/Common/Common下

也可以放在具体的模块下的Common下.

如 /APP/Home/Common下



## 10.6 自动填充

给Model定义\_auto属性,即可自动填充

```
array(  
    array(完成字段1,完成规则,[完成条件,附加规则]),  
    array(完成字段2,完成规则,[完成条件,附加规则]),  
    .....);
```

如果是对商品Model设定的填充规则

```
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.

用**insertFields** 和 **updateFields** 来限制允许插入和更新的字段.

例: 自动过滤,不在如下列表中列,将被过滤掉,不会Update相应字段

```
protected $updateFields = '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,last_update
```

## 10.8 商品列表页

```
class GoodsController extends Controller {  
    public function lists() {  
        // 查询商品  
        $model = new \Home\Model\GoodsModel();
```

```

$model->field('goods_id,goods_name,goods_sn,shop_price,is_sale,is_b
est,is_new,is_hot,goods_number');
$goods = $model->order('goods_id desc')->limit(0,10)->select();

//print_r($goods);
$this->assign('goods',$goods);
$this->display();
}

```

## 10.9 商品的删除

```

public function del() {
    $model = D('Home/Goods');
    $rs = $model->delete($_GET['goods_id']);
    if($rs) {
        $this->redirect('Admin/Goods/lists');
    } else {
        echo 'error';
    }
}

```

## 10.10 无限级分类[可直接复制]

```

// 查询指定栏目下的所有子孙栏目
public function __construct() {
    parent::__construct();
    $this->cats = $this->select();
}

public function getTree($id=0 , $lev=0) {
    $tree = array();

    foreach($this->cats as $v) {
        if($v['parent_id'] == $id) {
            $v['lev'] = $lev;
            $tree[] = $v;
            $tree = array_merge($tree , $this->getTree($v['cat_id'] , $
lev+1));
        }
    }

    return $tree;
}

```

# 第11章 TP功能库[要求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'))
```

调用示例:

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

// 重要,指定上传文件所在的根目录
$upload->rootPath = APP_PATH;

// 在Public下创建images/Y-m-d这样格式的子目录
$upload->savePath = '/Public/images/';

$info = $upload->upload();

if($info) {
    $model->ori_img = $info['xx']['savepath'] . $info['xx']['savename']
;
}
```

## 11.2 图片缩略

类:\Think\Image

调用流程:

```
// 1: 实例化
$img = new \Think\Image();

// 2: 打开图片
$img->open('原始图绝对路径');

// 3: 缩略保存
$img->thumb(宽,高,缩略选项)->save('保存路径');
```

## 11.3 分页功能

类:\Think\Page

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即为分页代码,显示到模板即可
```

- 注意: \*分页类已经计算出limit所需要的offset,N

商品页面实例:

```
$model->count('*'); // 计算总条数
$pager = new \Think\Page($cnt , 5);
$page = $pager->show();

$model->order('goods_id desc');
$model->limit($pager->firstRow,$pager->listRows)
$goods = $model->select();
```

## 11.4 验证码

类:\Think\Verify

```
class UserController extends Controller {
    public function vcode() {
        $code = new \Think\Verify(); // 实例化
        $code->imageW = 100; // 设置宽
        $code->imageH = 35; // 设置高
```



```
$code->fontSize = 14; // 设置字体
$code->length = 4;    // 设置验证码字数
$code->entry();       // 输出验证码
    }
}
```

调用如下:

```

```

验证办法:

```
$verify = new \Think\Verify();
$verify->check('XXXX');//即判断验证码是否为'XXXX'
```

## 第12章 完成前台页面

### 导入测试数据

把ecshop的测试数据导入商城,便于测试

shop.sql导入

图片放入/Public/images目录下

### 12.1 查询热门商品

```
// 查询精品商品
$bests = $model->where('is_best=1')->order('goods_id desc')->limit(3)->
select();
```

### 12.2 查询栏目下的商品

```

$cat_id = I('get.cat_id');
$goods = D('Home/Goods');

$list = $goods->field('goods_id,goods_name,goods_img,shop_price,market_price')->where(array('cat_id'=>$cat_id))->order('goods_id desc')->limit($page->firstRow , $page->listRows)->select();
$this->assign('list' , $list);
$this->display();

```

## 12.3 商品页面

```

$goods_id = I('get.goods_id');
$goods = D('Home/Goods');
$row = $goods->find($goods_id);

```

## 12.4 面包屑

```

/**
 * getFamily 获取上级,上上级,... 面包屑导航
 */
public function getFamily($cat_id) {
    $fm = array();

    while($cat_id > 0) {
        foreach ($this->cats as $c) {
            if($c['cat_id'] == $cat_id) {
                $fm[] = $c;
                $cat_id = $c['parent_id'];
                break;
            }
        }
    }

    return array_reverse($fm);
}

```

# 第13章 cookie/session处理

## 13.1 会员注册

```
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;
```

```
class UserModel extends Model {  
  // 设定自动验证  
  protected $_validate = array(  
    array('username', '/\w{6,16}/', '用户名6-16位, 数字字母下划线组成', '1',  
    'regex', 1),  
    array('email', 'email', 'email不合法', 1, '', 1),  
    array('password', '6,16', '密码在6-16位', 1, 'length', 1),  
    array('passwordcfm', 'password', '2次密码不一致', 1, 'confirm', 1),  
  );  
  
  public function reg() {  
    $this->encPass();  
    return $this->add();  
  }  
  
  /**  
   * 对明文密码加盐, 加密  
   * @return 加密后的md5密码  
   */  
  protected function encPass() {  
    $this->salt(); // 生成盐  
    $this->password = md5($this->password . $this->salt);  
  }  
  
  /**  
   * 创建用户的盐  
   */  
  protected function salt() {  
    if(!$this->salt) {  
      $this->salt = $this->randStr();  
    }  
  
    return $this->salt;
```

```

    }

    protected function randStr($length = 8) {
        $str = ' ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01
23456789' ;
        return substr(str_shuffle($str) , 0, $length);
    }

    // 检测用户名和密码,参数收到的是post来的明文密码
    /*
    find()之后:$this [username:lisi,password:'md5fdsafdsafdsfa' , salt:a
bcdeefg]
    重赋明文密码:[username:lisi,password:'123456' , salt:abcdeefg]
    重加密:$this[username:lisi,password:'md5kjlfsfksfsdf' , salt:abcdee
fg]
    */
    public function checkPass($password) {
        $selfpass = $this->password;
        $this->password = $password;

        return $selfpass === $this->encPass();
    }

    // 设置用户cookie,完成登陆
    public function auth() {
        cookie('user_id' , $this->user_id);
        cookie('username' , $this->username);
        cookie('ccode' , encCookie($this->user_id , $this->username));
        return ture;
    }

    // 对于UserController,如下调用
    /**
    * @file UserController.class.php
    */
    class UserController {
        protected $model = null;

        public function __construct() {
            parent::__construct();
        }

        public function zhuce() {
            if(!IS_POST) {
                $this->display('user_login');
            } else {
                $verify = new \Think\Verify();
                /*if(!$verify->check($_POST['vcode'])) {
                    echo 'vcode wrong!';
                    exit;
                }*/
            }
        }
    }

```



```

        if(!$this->model->create()) {
            echo $this->model->getError();
        } else {
            if($user_id = $this->model->reg()) { // 注册成功
                // 利用刚刚生成的user_id,查询user表,刷新model的数据
                $this->model->find($user_id);
                $this->model->auth(); // 用户登陆
                $this->redirect('Home/Index/index');

            } else {
                echo 'reg fail';
            }
        }
    }
}

```

用户注册时,对象的变化过程:



## 13.2 会员登陆/退出

```

/**
 * @file controller/UserController
 */
public function login() {
    $cond = array('name' => I('post.name'));
    $user = D('Home/User');
    if(!$user->where($cond)->find()) {
        echo '用户名不存在';
    } else if(!$user->checkPass(I('post.password'))) {
        echo '密码不对';
    } else {
        // 登陆正确
        $user->auth();
        $this->redirect('/');
    }
}

public function logout() {
    $this->model->revoke();
    $this->redirect('Home/User/login');
}

```

```
// file UserModel
// 授权
public function auth() {
    cookie('user_id' , $this->user_id);
    cookie('username' , $this->username);
}

// 收回权限
public function revoke() {
    cookie('user_id' , null);
    cookie('username' , null);
}
```

## 13.3 cookie安全加强

```
/**
 * 给用户分配cookie 授权
 */
public function auth() {
    cookie('user_id' , $this->user_id);
    cookie('username' , $this->username);
    cookie('ccode' , encCookie($this->user_id , $this->username));
}

/**
 * file APP/Common/Common/function.php
 * 加密cookie
 */
function encCookie($user_id,$username) {
    $ccode = md5($user_id . '|' . $username . '|' . C('COO_KEY'));
    return $ccode;
}

/**
 * file MODULE/Common/Conf/config.php
 */

// 配置文件中添加 本站的cookie密钥
'COO_KEY'          => 'JKD^&*&#%$JK0FD!@s3',
```

## 13.4 session最近浏览

```

/**
 * @file Controller/Indexcontroller.class.php
 */
public function goods {
    // ...省略无关逻辑,开始记录浏览商品...

    $this->history($row);
}

// 记录浏览历史
protected function history($row) {
    $history = session('?history') ? session('history') : array();

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

    $history[$row['goods_id']] = $g;

    if(count($history) > 5) {
        $key = key($history);
        unset($history[$key]);
    }

    session('history' , $history);
}

public function index() {
    // ....
    // 赋值浏览记录 , 注意,true是为了保证不改变数组的键
    $this->assign('history' , array_reverse(session('history') , true));
    // ....
}

```

## 第14章 第3方库的导入

### 14.1 导入foo测试类

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,发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();

也可以通 D('Home/Xxx','Tool');

## TP第15章 在线下单

### 15.1 订单表

```
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是到付  
  paystatus tinyint not null default 0,#0未支付,1已付  
  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;
```

#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;
```



## 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

### 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个商品的数量,如果减至0,则从购物车删除此商品
     * @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();

    /**
     * 返回购物车中有几种商品
     * @return int
     */
}
```

```

abstract public function calcType();

/**
 * 返回购物中商品的个数
 * @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 CartTool extends ACartTool{
    public static $ins = null;
    public $item = array();

    final protected function __construct() {
        if(session('?cart')) {
            $this->item = session('cart');
        }
    }

    public static function getIns() {
        if($ins === null) {
            self::$ins = new self();
        }

        return self::$ins;
    }

    /**
     * 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'=>$s

```

```

hop_price , 'num'=>1);
        $this->item[$goods_id] = $goods;
    }
}

/**
 * 减少购物车中1个商品的数量，如果减至0，则从购物车删除此商品
 * @param $goods_id int 商品id
 */
public function decr($goods_id) {
    if(isset($this->item[$goods_id])) {
        $this->item[$goods_id]['num'] -= 1;
    }

    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;
}

/**
 * 返回购物车中有几种商品
 * @return int
 */
public function calcType() {
    return count($this->items);
}

/**
 * 返回购物车中商品的个数
 * @return int
 */
public function calcCnt() {
    $cnt = 0;
    foreach($this->items as $v) {
        $cnt += $v['num'];
    }
}

```

```

        return $cnt;
    }

    /**
     * 返回购物车中商品的总价格
     * @return float
     */
    public function calcMoney() {
        $money = 0;
        foreach($this->items as $v) {
            $money += ($v['shop_price'] * $v['num']);
        }

        return $money;
    }

    public function clear(){
        $this->item = array();
    }

    public function __destruct() {
        session('cart' , $this->item);
    }
}

```

## 15.3 购物流程

```

/**
 * @file Controller/Flowcontroller.class.php
 */
/**
 * 把一个商品添加到购物车
 */
public function add() {
    $goods = D('Goods');
    if(!$goods->find(I('get.goods_id'))){
        $this->redirect('/');
        exit;
    }

    // 实例化购物车对象
    $cart = \Home\Tool\CartTool::getIns();
    // 把商品添加到购物车中
    $cart->add($goods->goods_id , $goods->goods_name , $goods->shop_price);

    $this->assign('items' , $cart->items());
}

```



```

$this->assign('money' , $cart->calcMoney());
// 输出购物车
$this->redirect('Home/Flow/checkout');
}

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

    // 写入ordinfo表
    $_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ordgoods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);
        $og->where(array('ordinfo_id'=>$ordinfo_id))->delete();
        $this->show('下单失败~' , 'utf-8');
        exit;
    }

    $this->assign('ord_sn' , $_POST['ord_sn']);
    $this->assign('money' , $cart->calcMoney());

    $CBPay = new \Home\Pay\CBPay();
    $CBPay->v_oid = $_POST['ord_sn'];
    $CBPay->v_amount = $cart->calcMoney();

```

```
$this->assign('pay' , $CBPay->form());

// 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/快钱/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,回调地址写在配置文件中
        $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_am
ount , $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个密钥,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;
```

### 17.2 继承关联模型

要使用TP的关联查询效果,需要继承自关联模型

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

### 17.3 声明关联关系

一对一,  
一对多,  
多对多

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.

声明如下关系:



```
protected $_link = array(
    'comment'=> self::HAS_MANY,
);
```

在controller如下调用:

```
$goods = D('/Home/Goods');
$goods->find(23); //查出23号商品
$comms = $goods->relationGet('comment'); // 查出多条相关的评价。
```

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.

## 第18章 编辑器

ckeditor (国外)

ueditor (百度)

## 第19章 URL美化

```
// 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,代表以下四种模式:
// 0 (普通模式); 1 (PATHINFO 模式); 2 (REWRITE 模式); 3 (兼容模式) 默认为PATHINFO 模式
'URL_MODEL' => 1,

// PATHINFO模式下,各参数之间的分割符号
'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

注: pathinfo是靠\$\_SERVER['PATH\_INFO']的值分析出参数

某些webServer,如nginx,默认没有pathinfo,

需要手动配置以支持pathinfo模式.(nginx课程)

- **2为 rewrite模式** URL例: /Home/Index/goods/goods\_id/34.html 注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#号,并重启apache
#LoadModule rewrite_module modules/mod_rewrite.so
```

注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3 上面的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

```
/*
file: Admin/Controller/CatController.class.php
method: add();
*/

// 原方法,每次调用getTree()查询数据库得到栏目树
$cat = D('Home/Cat');
$cats = $cat->getTree();

// 用缓存后,刷新2次看效果
$cats = S('cats');
if($cats === false) {
    echo 'from mysql';
    $cats = $cat->getTree();
    S('cats' , $cats ,300);
}
```

## 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

`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

`$model->cache(true)->where()->order()->xxxx`