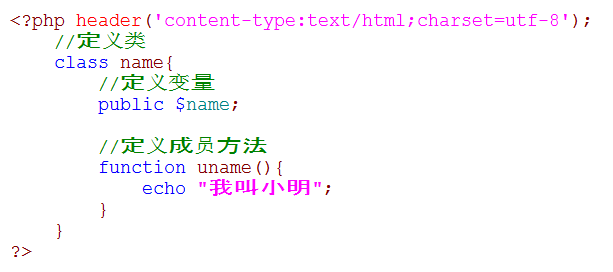
**PHP3面向对象（3粗，小4）**

**一 面向对象概念：**我们常说的面向对象编程 包括3部分内容：

* 面向对象分析（Object Oriented Analysis,OOA）
* 面向对象设计（Object Oriented Design,OOD）
* 面向对象编程（Object Oriented Programming,OOP）

**二 类和对象：**

* 类：属性和方法的集合（面向对象编程的核心和基础）；通过class关键字和类名进行定义。例如：

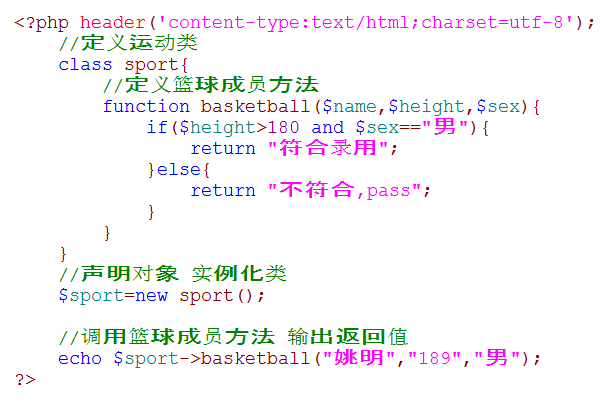


* 成员方法：类中的函数成为成员方法；
* 函数和成员方法的区别：函数实现某个独立功能；成员方法实现类的一个行为。
* 对象：类的实例化。

通过关键字new声明一个对象；

调用方法：对象名一>成员方法。

例如：



* 成员变量：类中的变量（属性/字段）

作用：保存数据信息；成员方法进行交互来实现某项功能；

调用方法：关键字 成员变量名；

关键字：

Public：公共的

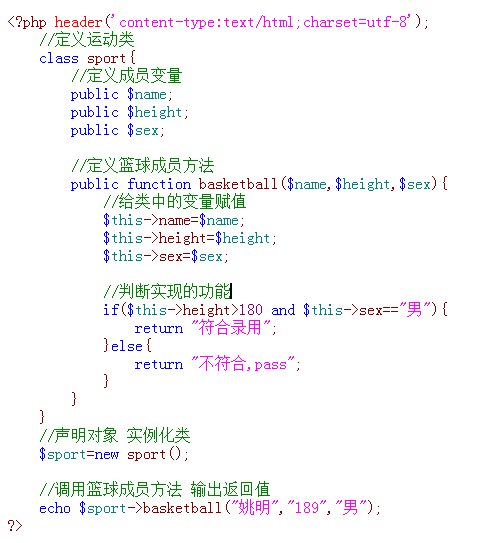
private：私有的

protected：受保护的

static：静态的

final：最终的

const：类常量



* $this🡪作用：调用本类中的成员变量，成员方法。用法：$this🡪name;后面变量没有$符号。

**三 面向对象编程的三大特点**

* 封装性：（隐藏性）将类的的使用和实现分开，只保留有限的接口与外部链接，避免了程序之间的相互依赖。
* 继承性：自动继承父类的属性和方法，可以重写或添加；简化对象和类的创建，提高代码的重用性；分为单继承和多继承，php只支持单继承（只有一个父类）。
* 多态性：同一个类的不同对象，调用同一个方法可以获得不同的结果；增强了软件的灵活性和重用性。

四

**第二单元：**

**第三单元：**

**Smarty模板技术**

**一 Smarty简介**

Smarty是PHP编写的模板引擎；优点实现了程序的分离，逻辑更加清晰，利与优化。(下载地址：http//smarty.php.net/download.php)

**二 Smarty与MVC**

Smarty这种开发模式，正是基于MVC框架概念。

MVC(Model-View-Controller,模型-视图-控制器)，是指一个应用程序由3部分构成：模型部分、视图部分和控制部分。

模型：对接收的信息进行处理，并将处理的结果传给视图。相当于PHP后台页面。

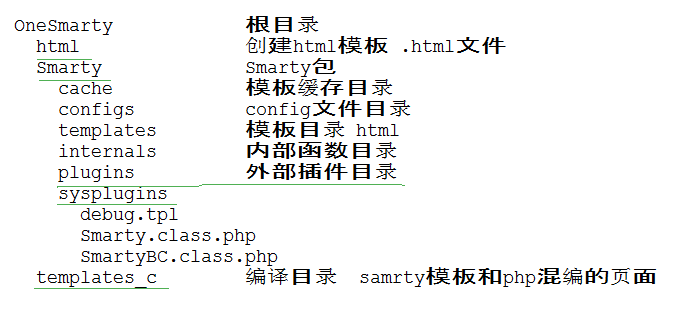
视图：就是提供给用户界面。相当于HTML页面。

控制：负责处理视图和模型的对应的关系。相当于Smarty代码。

**三 Smarty特点**

* 相对于其他模板速度快
* 可以自行设置模板定界符
* 仅对修改过的模板文件重新编译
* 模板中可以使用if/elseif/else/endif
* 内建缓存支持
* 可自定义插件

**四 搭建Smarty程序**

****

**五 使用Smarty加载一个页面**

****

**六 常用方法及相关细节**

* assign()方法。用于给模板赋值。

assign('变量名',值); 例如：$smarty->assign('data',$data);

* display()方法。用于显示模板。(编译号？)

display("表单页面"); 例如：$smarty->display("index.html");

* Smarty模板中所有注释，变量，函数等都要包含在定界符内。Html文件
* 注释：{\*这是注释\*}

**七 变量**

* PHP页面中的变量

Html页面用$+变量名接收($data)

输出：1.和php中一样通过下标输出($val.[u\_id])；2.通过键值获取数组元素($val.u\_id)

* 保留变量

相当于php中的预定义变量，Smarty模板中使用保留变量时不需要使用assign()方法传值，直接调用。

1. get，post，server，session，cookie，request相当于php中($\_GET,$\_POST,…….);
2. now：当前时间戳，等价于PHP中的time();
3. const修饰常量
4. config配置文件内容变量

* 从配置文件中读取数据

1 . {#title#}。

2 . 使用保留变量中的$smarty🡪config来调用配置文件。

**八 修饰变量**

{变量名 | 修饰方法：参数} 例如：{$str | upper | n12br}

常用方法：

capitalize(首字母大写)；

count\_characters(变量中字符串的个数。参数true，算空格；false)。

cat：“characters”(将cat中的字符串添加到指定字符串的后面)。

date\_format：“%Y-%M-%D”(格式化时间和日期和php中strftime()·数一样)。

lower：变量字符串小写。

Upper：将变量改为大写。

\*修饰时可以多个方法共同使用，方法之间用“|”,分割。

**九 流程控制**

1. 判断if…elseif…else

{if 条件语句1 }

语句1

{elseif 条件语句2 }

语句2

{else}

语句3

{/if}

常用符号：



1. foreach循环结构

{foreach $data as $val}

{val.u\_name;}

{/foreach}

1. section循环控制

**十 Smarty程序设计**

* Smarty程序功能主要分为两种：

1. 和smarty模板进行交互，如：assign(),display();
2. 配置Smarty参数，如：template\_dir,$config\_dir等。

* Smarty的配置变量

1. 只有一个常量：SMARTY\_DIR保存Smarty类库的完整路径。
2. $template\_dir:模板目录。用来存放smarty模板，后缀名一般为.html。
3. $compile\_dir:编译目录。默认目录template\_c目录。模板和php混编的页面。
4. $cache\_dir:缓存目录。存放缓存文件 caching开启，能直接读取smarty文件。
5. $config\_dir:配置目录。存放配置文件。
6. $debugging:调试变量。打开调试控制台：在(config.php)中将$smarty->debugging设为true即可。
7. $caching:缓存变量。开启缓存。只要前模板文件未被改动。Smarty就直接从缓存目录中读取缓存文件而不重新编译模板。

**THINKPHP，TP3.2编程**

官网提供了稳定版本的下载：<http://thinkphp.cn/down/framework.html>

1.类文件都是以.class.php为后缀,使用驼峰法命名，并且首字母大写，例如 DbMysql.class.php；

2.类的命名空间地址和所在的路径地址一致，例如 Home\Controller\UserController类所在的路径应该是 Application/Home/Controller/UserController.class.php；

3. 类名和文件名一致（包括上面说的大小写一致），例如 UserController类的文件命名是UserController.class.php， InfoModel类的文件名是InfoModel.class.php， 并且不同的类库的类命名有一定的规范；

4. 方法的命名使用驼峰法，并且首字母小写或者使用下划线“\_”，例如 getUserName，\_parseType，通常下划线开头的方法属于私有方法；

5. 属性的命名使用驼峰法，并且首字母小写或者使用下划线“\_”，例如 tableName、\_instance，通常下划线开头的属性属于私有属性；

6. 常量以大写字母和下划线命名，例如 HAS\_ONE和 MANY\_TO\_MANY；

7. 数据表和字段采用小写加下划线方式命名，并注意字段名不要以下划线开头，例如 think\_user 表和 user\_name字段是正确写法，类似 \_username 这样的数据表字段可能会被过滤。

模型类的命名规则是除去表前缀的数据表名称，采用驼峰法命名，并且首字母大写，然后加上模型层的名称（默认定义是Model），例如：

| **模型名** | **约定对应数据表（假设数据库的前缀定义是 think\_）** |
| --- | --- |
| UserModel | think\_user |
| UserTypeModel | think\_user\_type |

如果你的规则和上面的系统约定不符合，那么需要设置Model类的数据表名称属性，以确保能够找到对应的数据表。

class IndexController extends Controller {

public $userName;

public function loginIndex(){

echo "欢迎进入thinkphp";

}

# htmlspecialchars

将特殊字符转成 HTML 格式。

语法: **string htmlspecialchars(string string);**

返回值: 字符串

函数种类: 资料处理

|  |  |
| --- | --- |
| |  | | --- | |  | |
| 内容说明 |

本函数将特殊字符转成 HTML 的字符串格式 ( &....; )。最常用到的场合可能就是处理客户留言的留言版了。

* & (和) 转成 &amp;
* " (双引号) 转成 &quot;
* < (小于) 转成 &lt;
* > (大于) 转成 &gt;

此函数只转换上面的特殊字符，并不会全部转换成 HTML 所定的 ASCII 转换。

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | |  | |

# nl2br

将换行字符转成 <br>。

语法: **string nl2br(string string);**

返回值: 字符串

函数种类: 资料处理

|  |  |
| --- | --- |
| |  | | --- | |  | |
| 内容说明 |

本函数将换行字符转换成 HTML 换行的 <br> 指令。

模板替换：（页面的跳转情况） 网站 入口文件 模块 控制器 方法

当前页面:<http://localhost/php3/PhotoSys/index.php/Admin/Login/index>

1. \_\_URL\_\_(\_\_CONTROLLER\_\_)：跳转:

<http://localhost/php3/PhotoSys/index.php/Admin/Login>当前控制器的URL

1. \_\_ROOT\_\_:跳转: <http://localhost/php3/PhotoSys/> 当前网站。
2. \_\_ACTION\_\_:跳转: <http://localhost/php3/PhotoSys/index.php/Admin/Login/index> 当前操作。
3. \_\_APP\_\_:跳转：<http://localhost/php3/PhotoSys/index.php> 会替换成当前应用的URL地址
4. \_\_MODULE\_\_：跳转：<http://localhost/php3/PhotoSys/index.php/Admin>会替换成当前模块的URL地址
5. \_\_SELF\_\_：跳转：<http://localhost/php3/PhotoSys/index.php/Admin/Login> 会替换成当前的页面URL
6. \_\_PUBLIC\_\_：跳转：<http://localhost/php3/PhotoSys/Public/> 会被替换成当前网站的公共目录 通常是 /Public/

**TP中的类：**

**文件上传：**

$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->savePath='./Public/Uploads/'; // 设置附件上传目录

$upload->rootPath="./";// 上传文件入口路径

$info=$upload->upload();

if(!$info) {

// 上传错误提示错误信息

$this->error($upload->getError());

}else{// 上传成功

$this->success('上传成功！');

}

缩略图：

$image = new \Think\Image();

$image->open($path);

//避免文件名重复 当前时间戳加随机数命名+thumb.jpg

$thumb\_img=time().rand(10000,99999).'.thumb.'.$info['n\_file']['ext'];

//echo $thumb\_img;die;

//生成缩略大小 存放文件的路径

$image->thumb(150, 150)->save("./Public/Thumb/$thumb\_img");

//拼写上传的路径

$data['n\_file']="./Public/Thumb/$thumb\_img";

打印print\_r($info);信息如下：

****

**常量的输出：**

**欢迎{$Think.cookie.username}登录**