TP 跨模块调用操作方法（A方法与R方法）

**通过 A 方法跨模块调用操作**

A 方法用于实例化其他模块（当于 new 关键字），模块被实例化之后，就可以以对象的方式调用模块内的操作。

A('User') //调用当前项目的 User 模块

A('Admin://User') //调用 Admin 项目的 User 模块

A('Admin/User') //调用 Admin 分组的 User 模块

A('Admin://Tool/User') //调用 Admin 项目 Tool 分组的 User 模块

**通过 R 方法跨模块调用操作**

ThinkPHP 还提供了 R 方法，可直接调用其他模块的操作方法，将上面使用 A 方法的例子更改为 R 方法：

R 方法同样也支持跨分组调用和跨项目调用方法，语法如下：

R('<项目名://><分组名/>模块名/操作'<, array()>)

R('User/showName') //调用当前项目的 User 模块的 showName 方法

R('Admin://User/showName') //调用 Admin 项目 User 模块的 showName 方法

R('Admin/User/showName') //调用 Admin 分组 User 模块的 showName 方法

R('Admin://Tool/User/info') //调用 Admin 项目 Tool 分组的 User 模块的 info 方法

**R 方法接收参数**

R 方法也支持对调用的方法传入参数，因为实际上调动的操作，可能是需要传入参数的。

R 方法的第二个参数为数组，该数组将作为被调用操作的参数传入。如下面的例子所示：

R( 'User/showName',array(5) );

要传入多个参数，那么将 R 方法的 array() 参数数组依次定义多个元素即可。

**A 方法还是 R 方法？**

从上面的例子可以看出，A 方法和 R 方法都可以调用其他模块的操作，那是使用 A 方法好还是 R 方法好呢？在此建议如下：如果要使用其他模块内的多个方法，那么建议使用 A 方法，通过对象的方式调用该模块的不同方法，避免多次实例化对象；如果只需要使用其他模块内的其中一个方法，那么无疑 R 方法是最简洁的。

**U方法**

U方法用于完成对URL地址的组装，特点在于可以自动根据当前的URL模式和设置生成对应的URL地址，格式为：  
U('地址','参数','伪静态','是否跳转','显示域名');  
在模板中使用U方法而不是固定写死URL地址的好处在于，一旦你的环境变化或者参数设置改变，你不需要更改模板中的任何代码。在模板中的调用格式需要采用 {:U('地址', '参数'…)} 的方式

**I方法**

ThinkPHP 中 I() 是获取变量值，可以通过这个方法，对变量进行验证，从而保证了数据的可靠性。  
I(‘变量类型.变量名/修饰符’,[‘默认值’],[‘过滤方法’],[‘额外数据源’])

变量类型是指请求方式或者输入类型，包括：

| **变量类型** | **含义** |
| --- | --- |
| get | 获取GET参数 |
| post | 获取POST参数 |
| param | 自动判断请求类型获取GET、POST或者PUT参数 |
| request | 获取REQUEST 参数 |
| put | 获取PUT 参数 |
| session | 获取 $\_SESSION 参数 |
| cookie | 获取 $\_COOKIE 参数 |
| server | 获取 $\_SERVER 参数 |
| globals | 获取 $GLOBALS参数 |
| path | 获取 PATHINFO模式的URL参数\*\*（3.2.2新增）\*\* |
| data | 获取 其他类型的参数，需要配合额外数据源参数\*\*（3.2.2新增）\*\* |

可以使用的修饰符包括：

| **修饰符** | **作用** |
| --- | --- |
| s | 强制转换为字符串类型 |
| d | 强制转换为整形类型 |
| b | 强制转换为布尔类型 |
| a | 强制转换为数组类型 |
| f | 强制转换为浮点类型 |

注意：变量类型不区分大小写。  
变量名则严格区分大小写。  
默认值和过滤方法均属于可选参数。  
变量修饰符是3.2.3版本新增

I('get.');// 获取整个$\_GET 数组

由于param类型是I函数默认获取的变量类型，因此事实上param变量类型的写法可以简化为：

I('id'); // 等同于 I('param.id')

I('name'); // 等同于 I('param.name')

path类型变量可以用于获取URL参数（必须是PATHINFO模式参数有效，无论是GET还是POST方式都有效），例如： 当前访问URL地址是[http://serverName/index.php/New/2013/06/01](http://servername/index.php/New/2013/06/01)

echo I('path.1'); // 输出2013

echo I('path.2'); // 输出06

echo I('path.3'); // 输出01

TP中用ajax做分页

**1.先做一个方法xianshi();**

public function xianshi(){

$n = D("chinastates");//造chinastates表的对象

$shuliang = $n->count();//取出数据的总条数

$page = new \Home\libs\Page($shuliang,15);//将page文件类引入。()里面需要参数，第一个参数是数据的总数量，第二个是每页显示多少条数据。所以上面要先求出数量。

$xinxi = $page->fpage();

$attr = $n->limit($page->limit)->select();//查询出所有数据,limit(0,15)，需要修改Page.class.php文件中的第57行$this->limit = "LIMIT ".$this->setLimit();，将 "LIMIT ".去掉，如果不去掉的话，将会显示limit(limit(0,15))，度了1个limit。

$this->assign("shuju",$attr);//将查询出的数据都注入显示页面

$this->assign("xinxi",$xinxi);

$this->show();//调用显示方法在显示页面显示。

}

**TP中使用ajax表单验证**

<?php

namespace Home\Controller;

use Think\Controller;

class TestController extends Controller{

public function tianjia() {//添加方法，用来显示页面

$this->show();

}

public function test(){//ajax处理方法

$y = D("YongHu");

$arr = array(//动态验证就是需要在哪验证就在哪里写验证方法。

array("uid","require","用户名不能为空"),//讲验证的方法写在方法里面

);

if($y->validate($arr)->create()){//这里要先调用validate方法，然后将写的验证方法放到validate里面

$this->ajaxReturn("通过验证","eval");

}else {

$this->ajaxReturn($y->getError(),"eval");

}

}

}

**[PHP中变量类型的判断](https://www.cnblogs.com/mood-butterfly/p/6439119.html)**

一、gettype()

二、php自带的类型判断如下

isset(): //变量是否已经声明

empty(): //变量是否为空

defined():　　　　　　//常量是否已经定义 define()

array\_key\_exists(mixed key, array search): //检查给定的键名或索引是否存在于数组中

is\_numeric ( mixed var ): //检测变量是 否为数字或数字字符串

is\_bool()、is\_float()、is\_int()、is\_string()、is\_object()、is\_array()、is\_null()

不难看出只要变量是否为” “或者0，或者是false和null，只要是这些值empty都会返回true，而isset是判断变量是否存在，只要你这个变量不是null或未赋值，返回结果都是true，而is\_null正好是isset的反结果。

**isset判断是否分配实际内存，empty判断是否有值且不为0。**

相对路径与绝对路径

未确定路径，凡是不以 . 或者 / 开头、也不是windows下 盘符:/ 开头的路径，例如:

a/a.php

common.inc.php

首先在逐一用include\_path中定义的包含目录来拼接[未确定路径]，找到存在的文件则包含成功退出，如果没有找到，则用执行require语句的php文件所在目录来拼接[未确定路径]组成的全路径去查找该文件，如果文件存在则包含成功退出，否则表示包含文件不存在，出错。 未确定路径比较容易搞混不建议使用。

**解决方案**

由于“相对路径”中的“参照目录”是执行入口文件所在目录，“未确定”路径也比较容易混淆，因此最好的解决方法是使用“绝对路径”; 例如b.php的内容如下，无论在哪里require b.php都是以b.php的路径为参照来require c.php的

$dir = dirname(\_\_FILE\_\_);  
require($dir . '../c.php');  
或者定义一个通用函数 import.php，将其设置为“自动提前引入文件”，在php.ini做如下配置

更改配置项（必须）auto\_prepend\_file = "C:\xampp\htdocs\auto\_prepend\_file.php"  
更改配置项（可选）allow\_url\_include = On

import.php内容如下

function import($path) {

$old\_dir = getcwd(); // 保存原“参照目录”

chdir(dirname(\_\_FILE\_\_)); // 将“参照目录”更改为当前脚本的绝对路径

require\_once($path);

chdir($old\_dir); // 改回原“参照目录”

}

这样就可以使用import()函数来require文件了，无论包含多少级“参照目录”都是当前文件。

PHP中获取url的内容

$content = file\_get\_contents("http://www.52fhy.com")； //将整个文件读入一个字符串

$lines = file("http://www.52fhy.com")； //把整个文件读入一个数组中

readfile("http://www.52fhy.com")； //读取文件并写入到输出缓冲

[实现页面跳转的几种方式](https://www.cnblogs.com/du892294464/p/6835277.html)

**在PHP脚本代码中实现**

1. header("location:helloworld.php"); //**在用header前不能有任何的输出。**页面会立即跳转，因为header执行了location重定向

   延迟跳转（比如登陆成功后会有几秒钟等待时间，然后跳转到了其他页面）

(2)header("Refresh:3;url=helloworld.php"); //会在3秒后执行跳转  
(3)sleep(3); header("location:url地址"); //调用了sleep()方法,效果也是x秒后执行跳转

**在js脚本代码中实现**  
1.window.location.href方法

<script type="text/javascript">

window.location.href="helloworld.php"

</script>

**使用js方法实现延迟跳转**

<script type="text/javascript">

setTimeout("window.location.href='helloworld.php'",3000);

</script>

2.window.location.assign方法  延迟跳转方法同上

<script type="text/javascript">window.location.assign("helloworld.php"）;

</script>

3.window.location.replace方法  (让新页面替换掉当前页面，不会保存在历史记录里，所有不能使用浏览器后退到原页面了)

<script type="text/javascript">

window.location.replace("helloworld.php");

</script>

4.window.open方法 三个参数，第一个URL地址。第二个打开新页面方式（比如新页面\_blank,\_new,自身跳转\_self）,第三个是新页面的方式，包括样式，位置等。

<script type="text/javascript">

window.open("index.php",\_blank,width=300px);

</script>

**使用HTML脚本代码完成跳转**

在<head>标签里执行代码，直接插入这句代码就可以

<meta http-equiv="refresh" content="3;url='helloworld.php'">