**前后台交互—表单应用**

目录

[二、 GET方式传值 4](#_Toc495592710)

[1. 表单方式 5](#_Toc495592711)

[1) 表单提交样式 5](#_Toc495592712)

[2) 数据提交格式 5](#_Toc495592713)

[2. 链接方式 6](#_Toc495592714)

[1) html的a标记： 6](#_Toc495592715)

[2) location.href属性 7](#_Toc495592716)

[3) location.assign()方法 7](#_Toc495592717)

[4) header的location与refresh选项 8](#_Toc495592718)

[三、 POST方式 9](#_Toc495592719)

[1. 表单设计 9](#_Toc495592720)

[2. 接收数据 10](#_Toc495592721)

[3. POST的使用场景： 10](#_Toc495592722)

[四、 PHP接收数据的三种方式 10](#_Toc495592723)

[1. 预定义变量 10](#_Toc495592724)

[1) 常见的预定义变量 10](#_Toc495592725)

[2) 显示服务器的配置信息 12](#_Toc495592726)

[2. $\_GET数组 13](#_Toc495592727)

[1) 注意事项 14](#_Toc495592728)

[3. $\_POST 14](#_Toc495592729)

[1) 注意事项 14](#_Toc495592730)

[4. $\_REQUEST 14](#_Toc495592731)

[1) 注意事项 15](#_Toc495592732)

[5. POST/GET方式的比较 15](#_Toc495592733)

[1) GET方式 15](#_Toc495592734)

[2) POST方式 15](#_Toc495592735)

[五、 PHP接收表单单选、复选框数据 16](#_Toc495592736)

[1. 下拉菜单的数据接收 16](#_Toc495592737)

[1) 表单域设置的注意事项 16](#_Toc495592738)

[2) 数据回显时的默认选中 16](#_Toc495592739)

[2. 单选项的数据接收 18](#_Toc495592740)

[1) 表单域设置的注意事项 18](#_Toc495592741)

[2) 数据回显时的默认选中 18](#_Toc495592742)

[3. 复选框的数据接收 19](#_Toc495592743)

[1) 表单注意事项 19](#_Toc495592744)

[六、 应用：学生信息注册 19](#_Toc495592745)

[1. 数据表设计 20](#_Toc495592746)

[1) 创建班级信息表 20](#_Toc495592747)

[2) 创建学生表，并录入学生 20](#_Toc495592748)

[2. 表单设计 20](#_Toc495592749)

[3. 数据接收 21](#_Toc495592750)

[1) 判断接收数据 21](#_Toc495592751)

[2) 使用数据，构建SQL 21](#_Toc495592752)

[七、 单文件上传 21](#_Toc495592753)

[1. 表单注意事项 21](#_Toc495592754)

[2. $\_FILES数组详解 22](#_Toc495592755)

[1) $\_FILES[‘myFile’][‘name’] 23](#_Toc495592756)

[2) $\_FILES[‘myFile’][‘tmp\_name’] 23](#_Toc495592757)

[3) $\_FILES[‘myFile’][‘type’] 23](#_Toc495592758)

[4) $\_FILES[‘myFile’]’[‘error’] 24](#_Toc495592759)

[5) $\_FILES[‘myFile’][‘size’] 24](#_Toc495592760)

[3. error错误代码含义 24](#_Toc495592761)

[4. 文件上传的完整步骤 26](#_Toc495592762)

[5. 文件上传入门案例： 27](#_Toc495592763)

[1) 设计表单 27](#_Toc495592764)

[2) 判断数组是否为空 27](#_Toc495592765)

[3) 错误提示 27](#_Toc495592766)

[4) 判断是否是合法的HTTP post上传文件 28](#_Toc495592767)

[5) 移动至永久目录 28](#_Toc495592768)

[八、 文件上传的相关问题 28](#_Toc495592769)

[1. 文件扩展名的简单获取方法 28](#_Toc495592770)

[2. 避免文件名重复 29](#_Toc495592771)

[3. 大小限定 29](#_Toc495592772)

[4. 文件类型限定（MIME类型） 29](#_Toc495592773)

[1) 获取文件MIME类型步骤 30](#_Toc495592774)

[九、 多文件上传 31](#_Toc495592775)

[1. 设计表单 31](#_Toc495592776)

[2. 接收数据 31](#_Toc495592777)

[3. 获取文件的完整信息 32](#_Toc495592778)

[1) 判断文件是否发生错误 32](#_Toc495592779)

[2) 获取每个文件的完整信息 33](#_Toc495592780)

[3) 处理单个文件上传 34](#_Toc495592781)

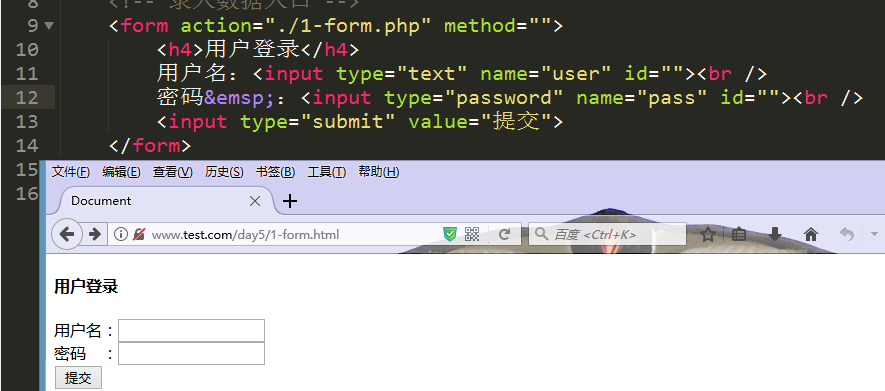
**回顾：**



**思考：**

1. **表单传值的两种方式是什么？**
2. **PHP接收表单数据的三种方式？**
3. **常见的输入类型数据如何接收？表单设计有哪些注意事项？**
4. **文件上传表单如何设计？如何将文件上传到服务器的根目录？如何判断文件的大小、类型，以及避免文件重名？**
5. **如何实现多文件上传？**

# GET方式传值



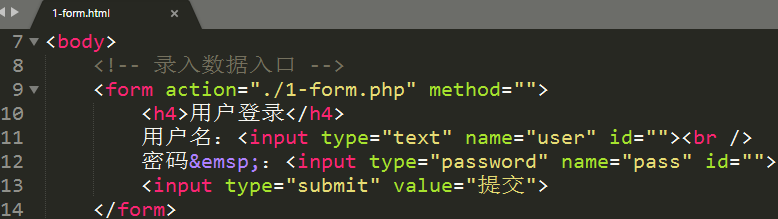
表单：提供用户向web服务器发送数据的接口。如：用户注册，用户登录，发布文章，发布产品信息等。

传值方式一般通过HTML中form表单的method指定。分为：**GET方式**（默认提交方式）、**POST方式**。此外还可以通过URL方式传值（本质还是GET方式）

## 表单方式

form 表单的METHOD设置为GET。表单的每个元素name属性值必须填写，否则该元素的值不提交。

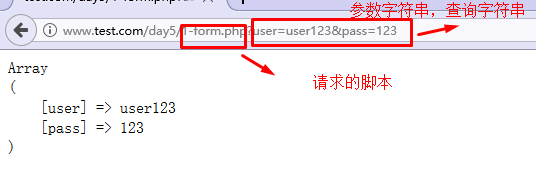
### 表单提交样式



### 数据提交格式

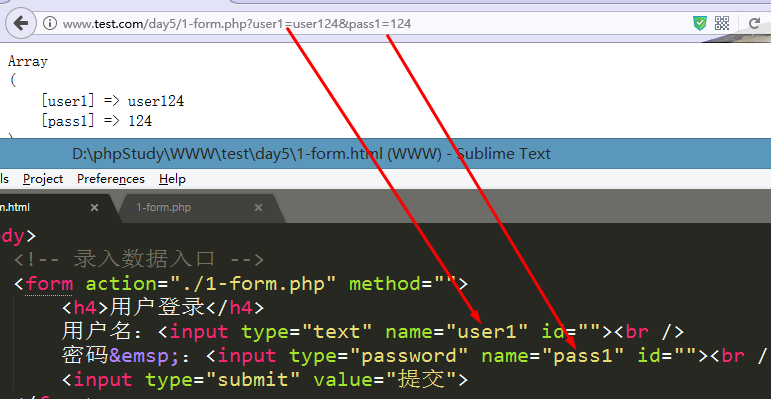
[www.eg.com/a.php?aaa=123&bbb=234](http://www.eg.com/a.php?aaa=123&bbb=234)





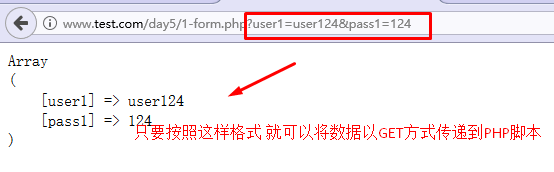
通过地址栏提交，数据被存放在$\_GET数组中。

注意form中各个输入元素的name属性值必填。



## 链接方式

只要在请求的URL中包含参数，参数就会以GET方式传递到请求的页面。



### html的a标记：

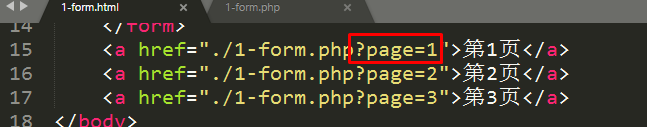
a href:引用地址

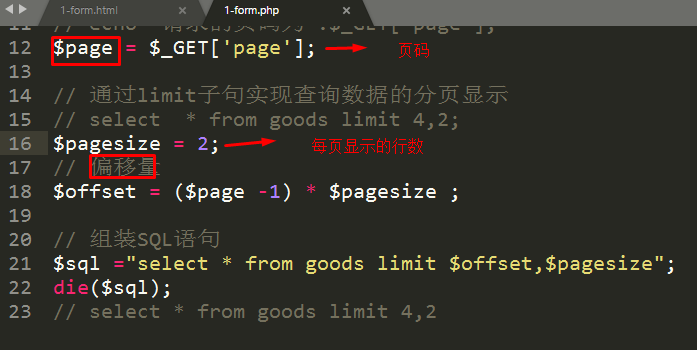
格式如下：

<a href=’请求的地址?a=100&b=200’>点击提交</a>

在请求的地址后使用?分隔，然后是查询字符串。

* 参数名与参数值之间没有空格。
* 参数值不需要使用单双引号包括。

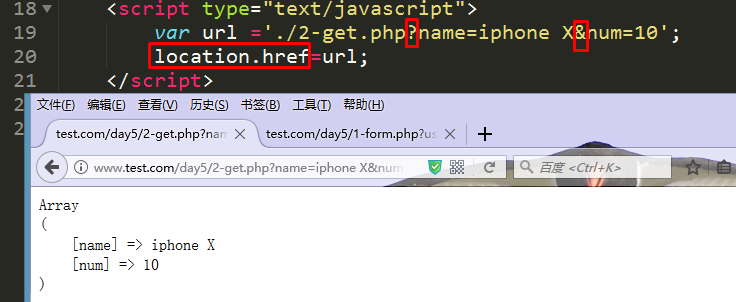




### location.href属性

location为JS对象。每个对象有属性（特点）、方法（能做的事情，函数）。JS跳转传递参数的本质还是在URL中包含参数。此时参数以GET方式传递到请求的页面。

location.href =”请求的地址?a=100&b=200”

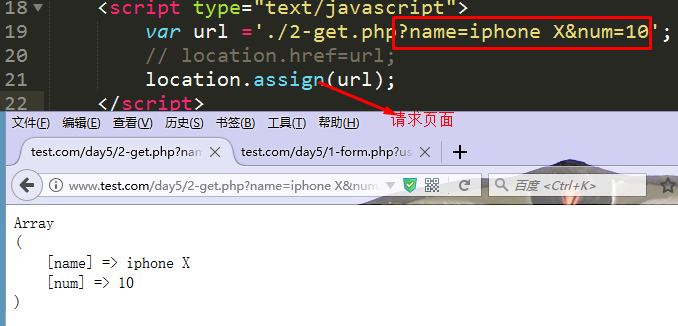


### location.assign()方法

Math.random()：生成随机数

方法的即对象的函数，实现特定的功能。

location.assign(url);



### header的location与refresh选项



header函数实现跳转以及刷新时，可以指定跳转的地址。如果在地址中添加对应的参数，则参数会被以GET方式提交到新的页面。

header(“location:跳转的地址”);

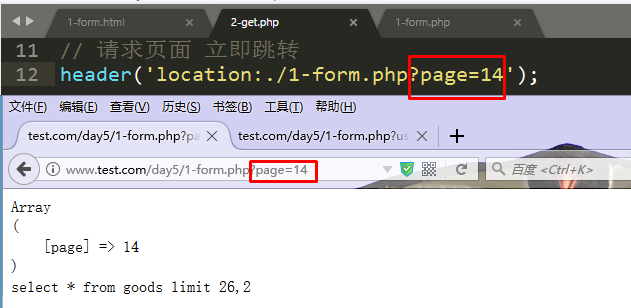
或者

header(“refresh:秒数；url=新的地址”)

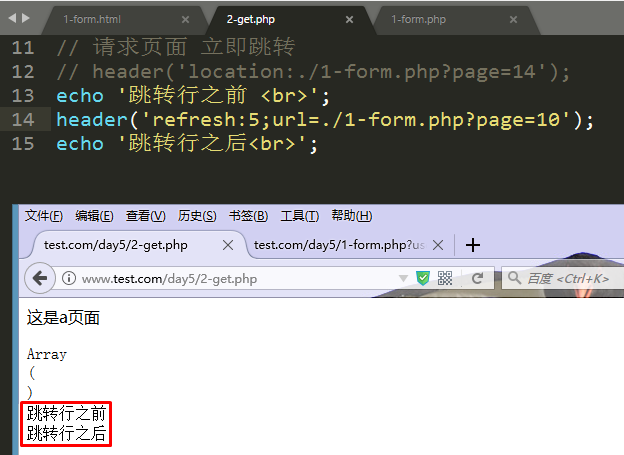
**两者的区别是：**

refresh方式，在等待时间结束之后，才跳转。location方式为直接跳转。

**location方式：**



**refresh方式：**



# POST方式

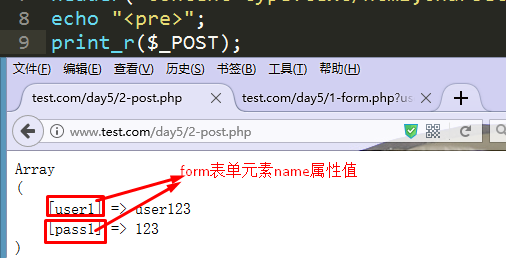
GET方式提交数据通过URL传递到请求的页面，提交的数据量一般较小。此外，GET方式传递的数据安全性低，对于密码等隐私信息、文件上传必须使用POST方式提交。

## 表单设计

要求：表单的method必须设置为POST。各个表单元素的name属性必填。

数据被提交到PHP脚本中$\_POST数组中。





## 接收数据

name属性必须添加属性值，否则在$\_POST数组中无对应的元素。（name属性值作为$\_POST关联数组元素的下标使用）



## POST的使用场景：

* 密码等安全性要求比较高的场合
* 提交的数据量比较大：发布文章，上传文件。

# PHP接收数据的三种方式

GET方式、POST方式提交的数据分别存放在$\_GET、$\_POST数组中。$\_GET与$\_POST数组都是系统预定义变量，关联数组，用户指定的参数名被作为元素的下标使用。

## 预定义变量

预定义变量：系统预先定义好的变量，可以直接使用的变量。

### 常见的预定义变量

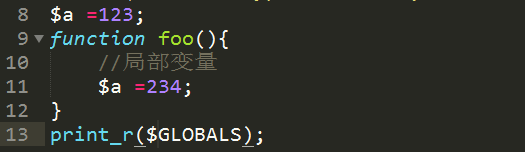
特点：变量名几乎是以’\_’开头，后半部分为全大写，都是数组。

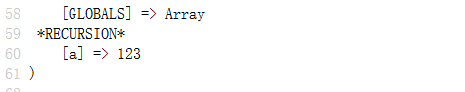


* **$\_SERVER:**存储服务器环境信息以及HTTP协议相关信息。
* $\_GET: 用户GET方式提交的数据。
* $\_POST: 用户POST方式提交的数据。
* $\_REQUEST: GET、POST、COOKIE数据。
* $\_COOKIE: 保存COOKIE数据（保存在浏览器端的用户隐私数据）。



* $\_SESSION: 保存SESSION数据（保存在服务器端的用户隐私数据）。
* $GLOBALS: 引用所有的全局变量数据。可以用来解决函数使用变量时的作用域问题（全局、局部 两者不通。）

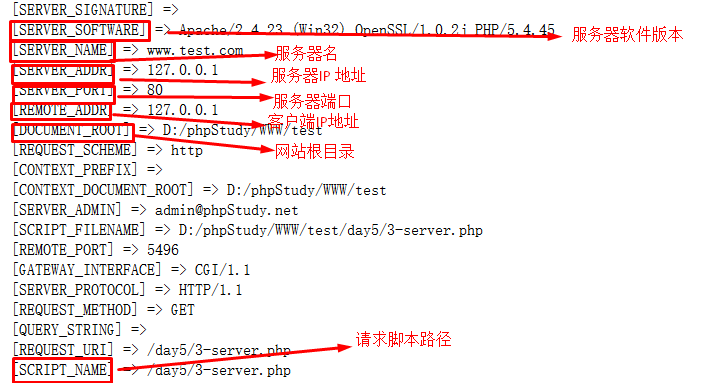




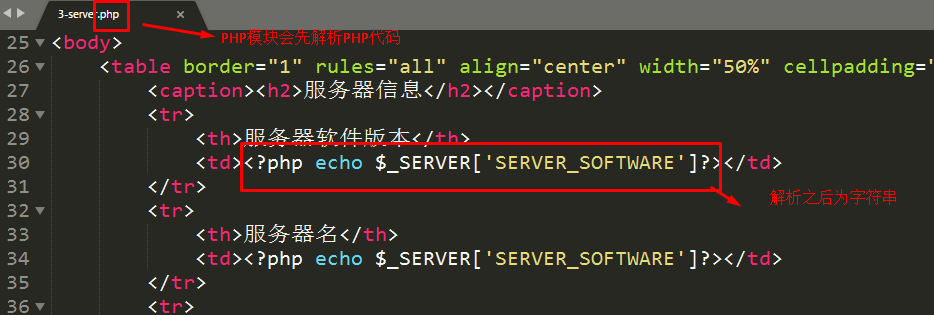
* $\_FILES: 上传的文件信息（包括文件名、文件类型、大小）

### 显示服务器的配置信息

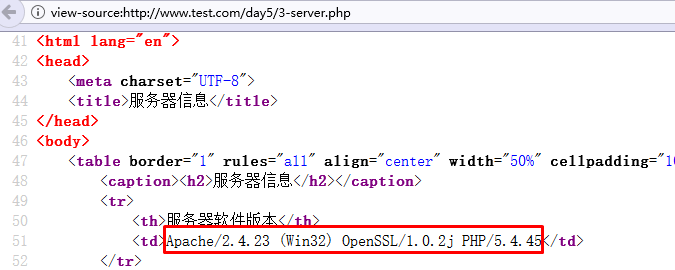
以表格形式展示服务器软件版本，服务器名，服务器端IP地址，端口号，客户端IP地址等



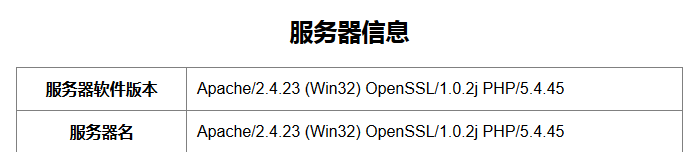
**初步效果：**



**页面源代码：**



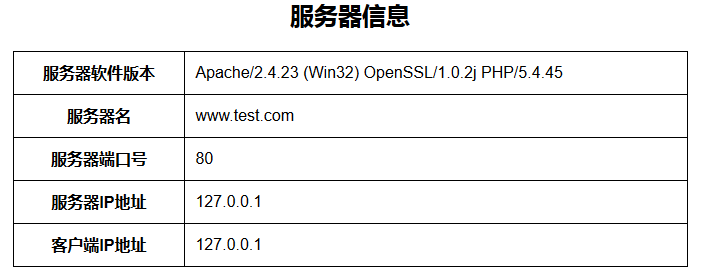
**执行效果：**



**完善：**



**最终效果：**



## $\_GET数组

关联-🡪下标从哪里来

### 注意事项

查询字符串（query\_string）的格式为

参数1=值1&参数2=值2

参数名与值之间不能添加空格，所有的提交数据都会以字符串类型保存在$\_GET数组中。

## $\_POST

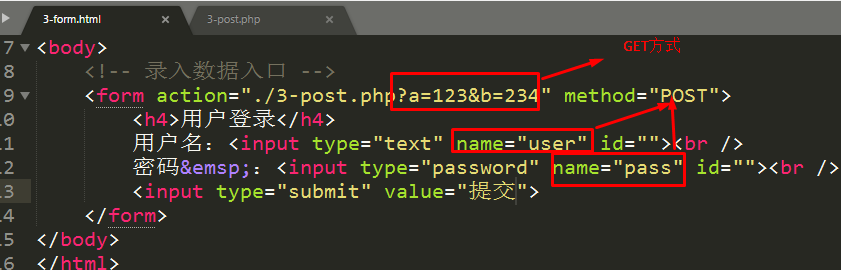
对表单数据安全性要求比较高、数据量比较大时选择POST方式提交。

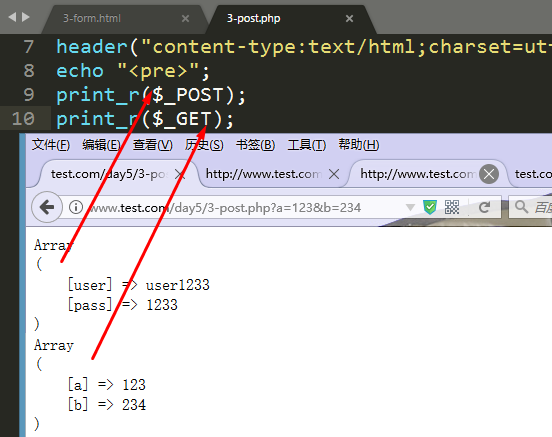
### 注意事项

form需要填写method=”POST”，此外表单元素的name属性值必须填写。

## $\_REQUEST

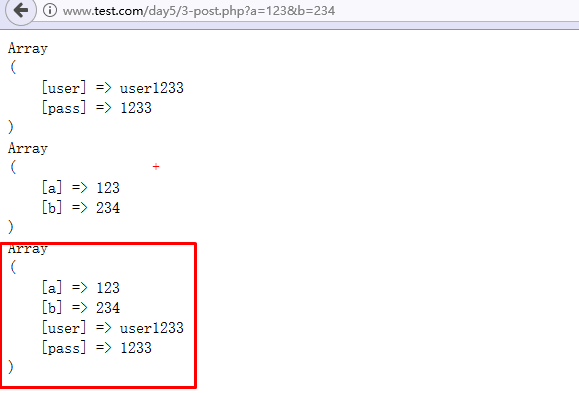
既提交GET数据、又提交POST数据，必须使用两个数组接收：





### 注意事项

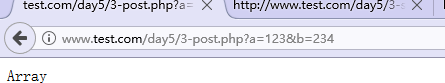
* 表单提交数据方式设置为POST方式后，通过action属性设置的提交方式仍为GET方式。
* $\_REQUEST =$\_GET + $\_POST +$\_COOKIE。Cookie中存储的一般为浏览器端保存的用户名及密码等信息。



## POST/GET方式的比较

### GET方式

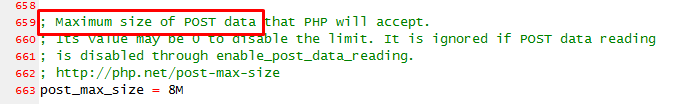
* 数据附在URL中提交，在地址栏中可见，此外在历史记录中也可以查看到提交的参数，**安全性低。**



* 数据上限一般为2kb,数据量小
* 由于直接包含参数，便于添加为书签(参数如页码，或者产品编号)。

### POST方式

* 不通过地址栏提交，数据相对安全。
* POST方式提交数据上限默认为8M（可以在PHP的配置文件post\_max\_size选项中修改）



# PHP接收表单单选、复选框数据

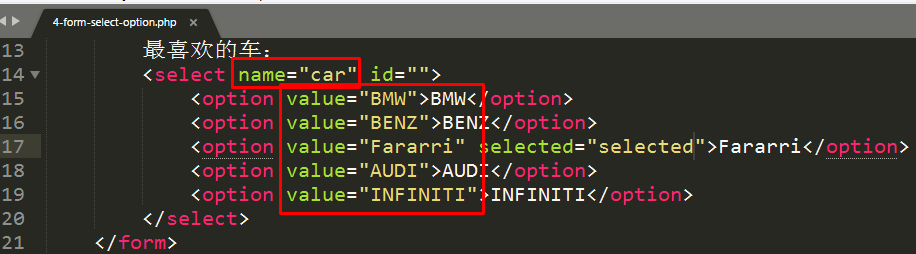
常见的输入类型：文本域（type=text）、单选按钮(type=radio)、多选按钮（复选项 type=checkbox）、下拉菜单(select>option)、多行文本(textarea)

## 下拉菜单的数据接收

### 表单域设置的注意事项

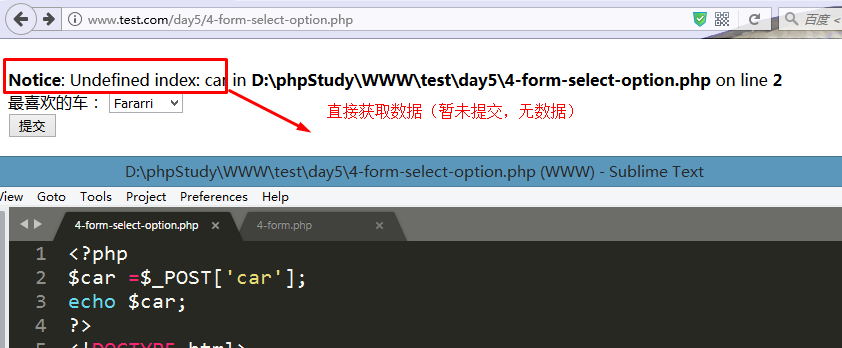
select >option实现的是单选项，下拉菜单。一般用于选项比较多的情况，如城市，省份，产品类型等等。

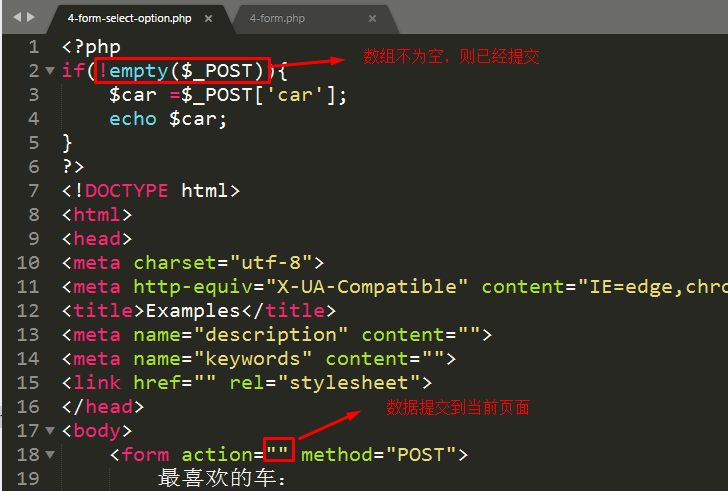
name属性必选。value值为必选。



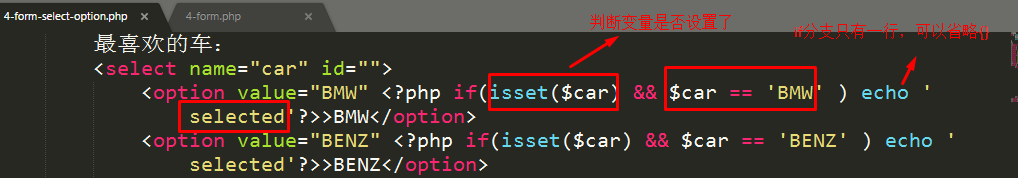
### 数据回显时的默认选中

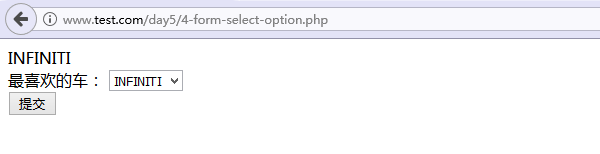
模拟修改页面的数据回显。



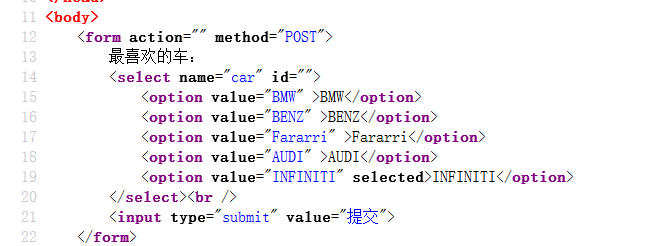


**用户选择的选项，提交后默认选中：**





源代码：

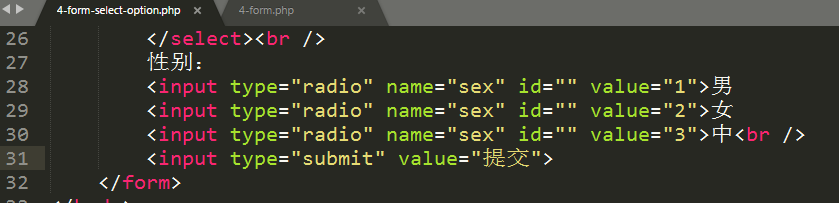


## 单选项的数据接收

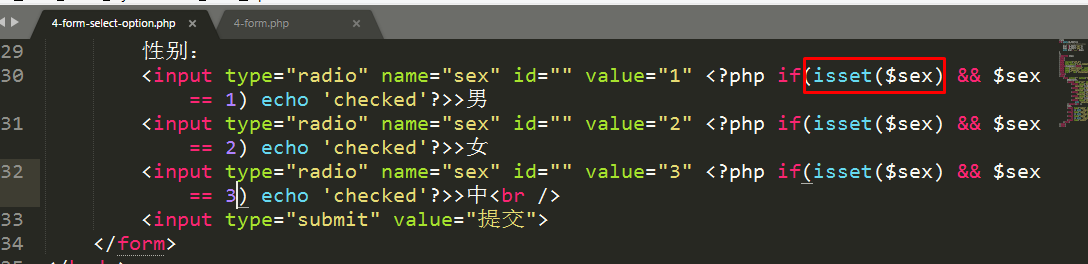
### 表单域设置的注意事项

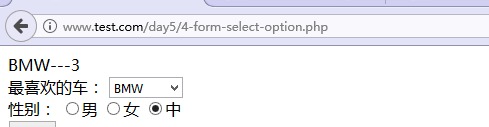
input标记的**type =radio**，单选按钮。

* 各个选项均需要指定name及value值。
* name值相同，value值不同。
* 可以使用checked设置默认选项



### 数据回显时的默认选中



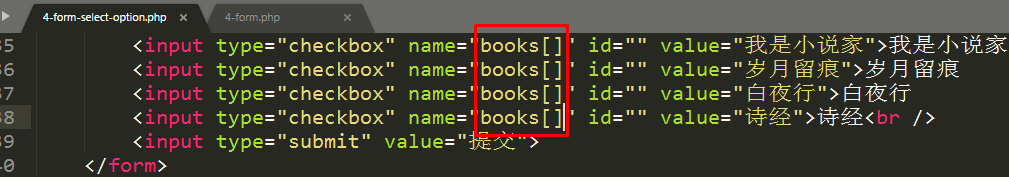


## 复选框的数据接收

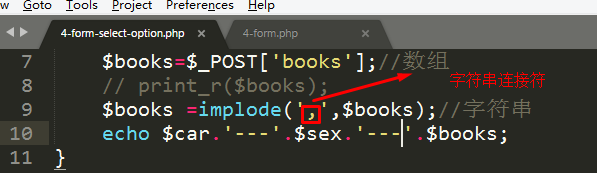
input的**type =checkbox,**可以同时选择多个选项。

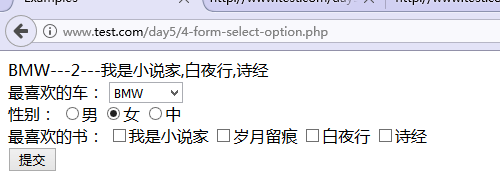
### 表单注意事项

* name命名形式必须为：XXX[]，最终数据才能将各个选项的值同时提交，否则只能提交最后一个勾选的属性值。不同的选项值，以数组元素的形式提交。
* value必选
* checked可选





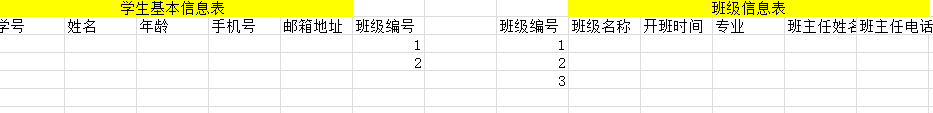




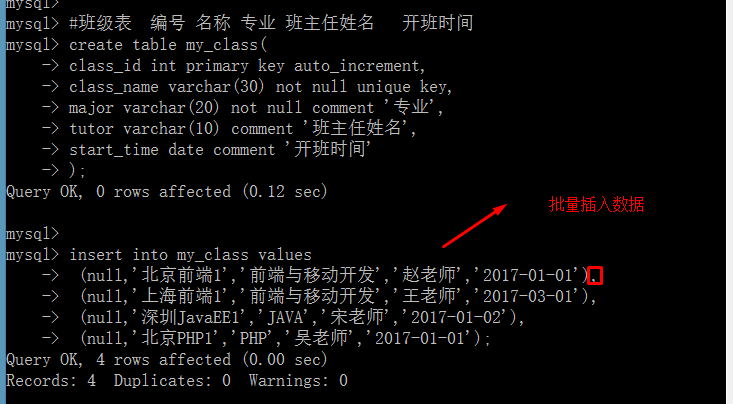
# 应用：学生信息注册

传智播客黑马程序员在全国各个小区开设各个专业、诸多班级。现在系统中录入学生，包括学生的学号、姓名，班级，手机号、邮箱等信息。

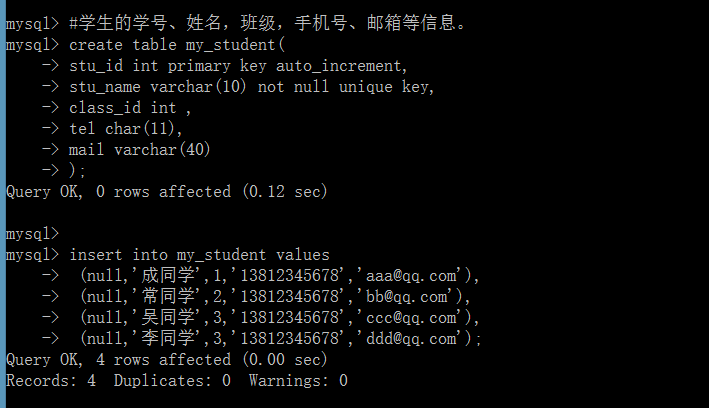
## 数据表设计



### 创建班级信息表

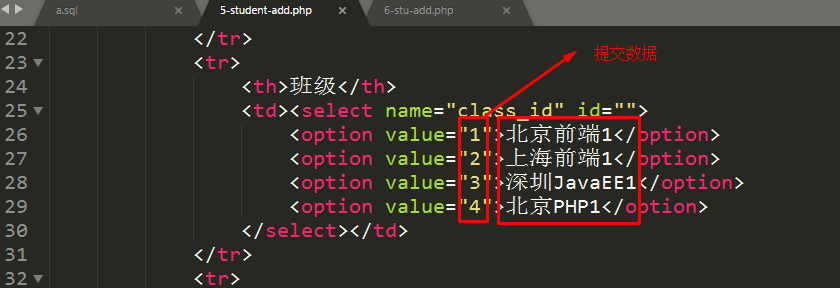


### 创建学生表，并录入学生



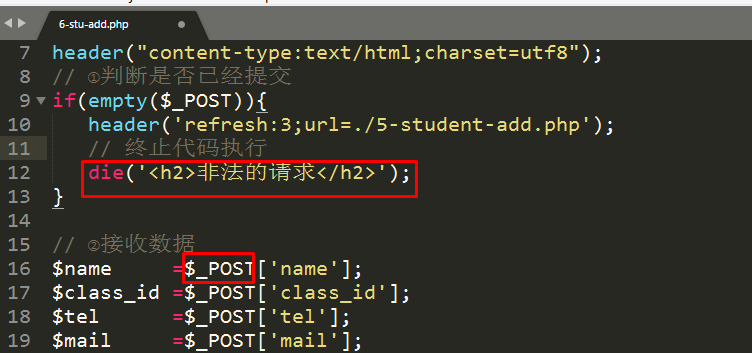
## 表单设计

设计数据表时，需要将数据库中存储的班级信息展示出来。暂时先使用固定数据填充。

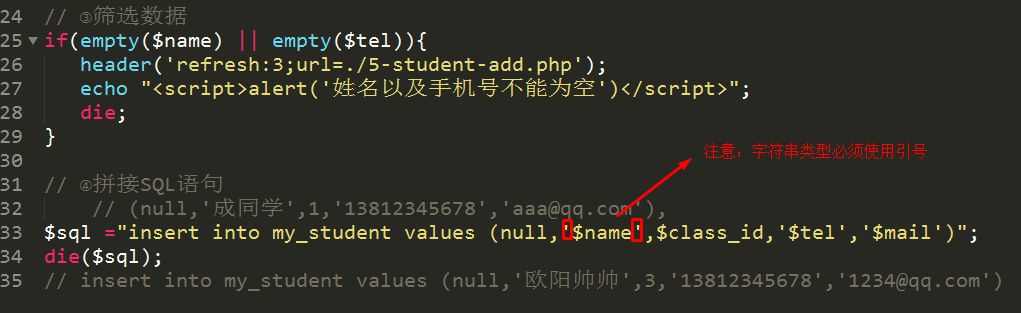


## 数据接收

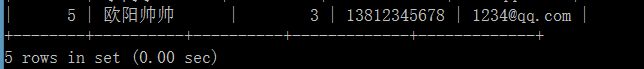
### 判断接收数据



### 使用数据，构建SQL



注意检查SQL语句的正确性。

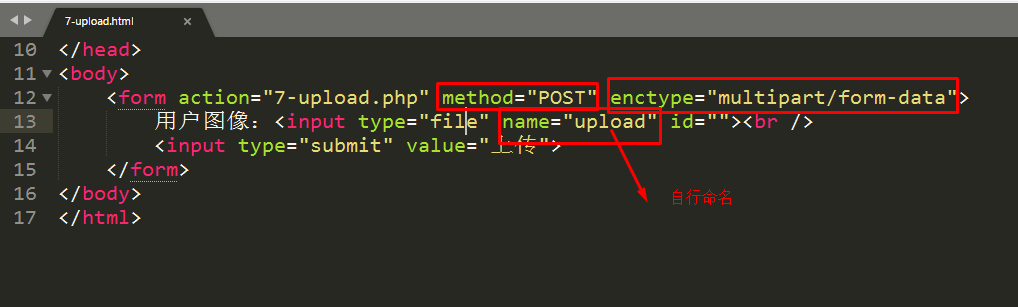


# 单文件上传

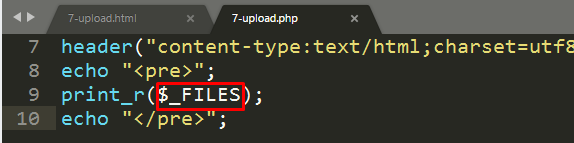
应用场景：添加文章或者商品时上传对应的图片信息。

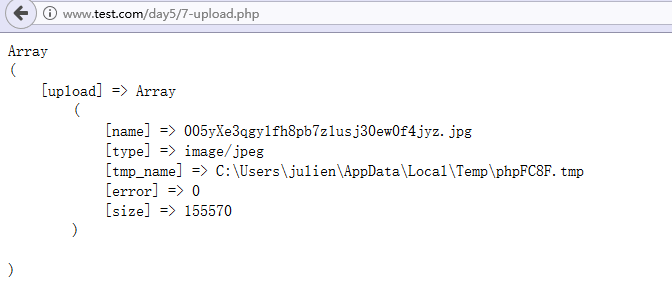
## 表单注意事项

* form 表单数据的编码类型由enctype（encode type）决定，默认为application/x-www-form-urlencoded方式编码。文件上传时属性值必须设置为multipart/form-data。
* 数据提交方式：POST方式
* 文件域：<input type=”file” name=”属性值”/>



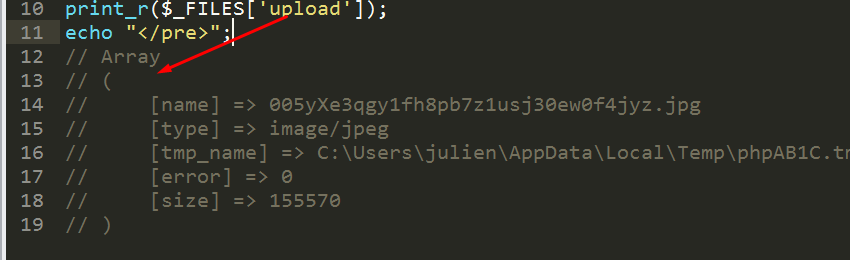
**数据接收：**





## $\_FILES数组详解

文件上传后，文件信息自动保存在预定义变量$\_FILES二维数组中。假设myFile为文件域的name属性值。



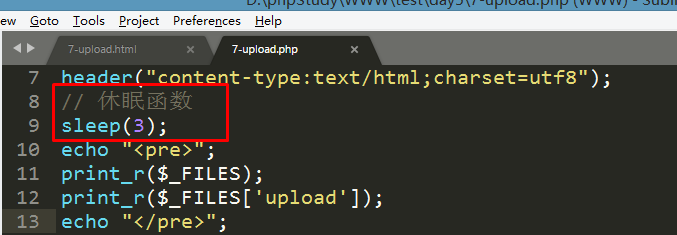
### $\_FILES[‘myFile’][‘name’]

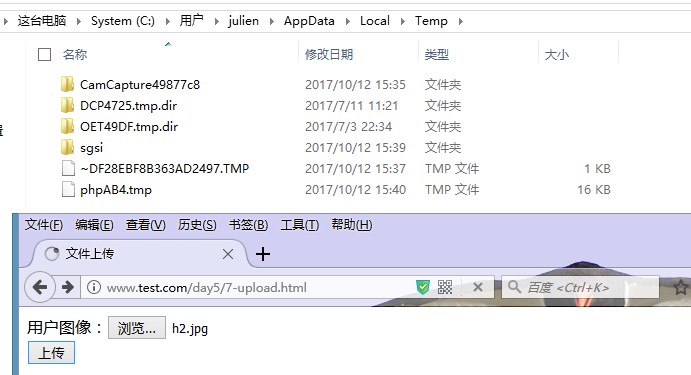
文件的原始名称。

### $\_FILES[‘myFile’][‘tmp\_name’]

文件的临时路径。没有直接将文件上传至Apache服务器的根目录，原因为：用户的数据是不安全。

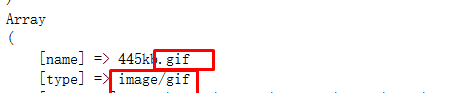
临时文件的查看方法：使用sleep函数，休眠代码的执行。





### $\_FILES[‘myFile’][‘type’]

文件的内容类型。



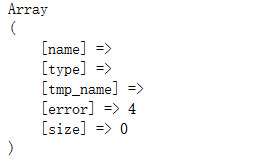
常见的文件类型：

* + 图像：image/jpeg,image/png,image/gif,
  + HTML:text/html HTML格式文件
  + JS文件：text/javascript

### $\_FILES[‘myFile’]’[‘error’]

文件上传的错误代号。0代表没有发生错误。

4：未选择文件



### $\_FILES[‘myFile’][‘size’]

文件大小（单位为byte，字节）

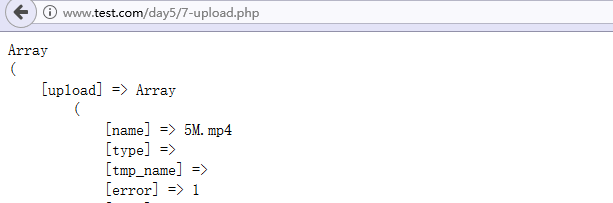
1M =1024 KB

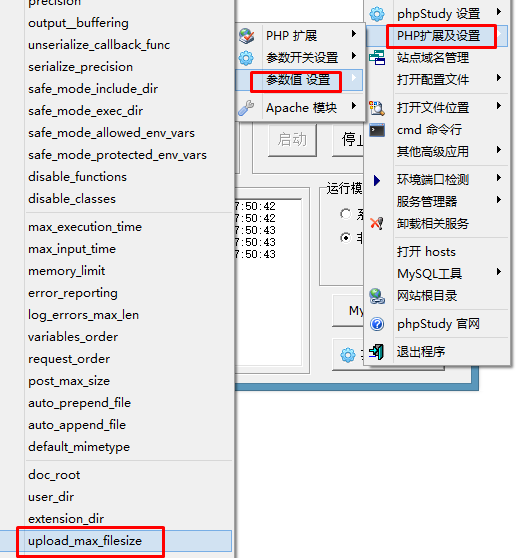
1kb =1024byte

1byte =8bit位

## error错误代码含义

* 0：文件上传至临时路径成功（将来需要将文件由临时路径移动到永久目录）
* 1：上传的文件超过了 php.ini 中 [upload\_max\_filesize](mk:@MSITStore:E:\DocRef\WEB开发必备参考资料\php手册2015.chm::/res/ini.core.html) 选项（默认为2M）限制的值

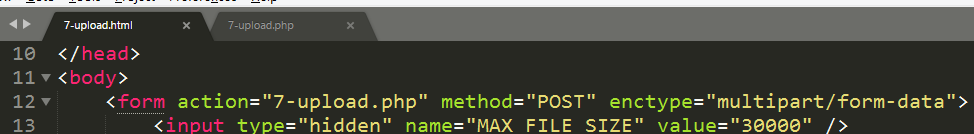


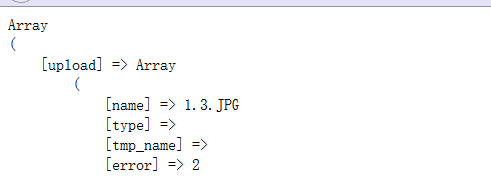


2:上传的文件大小超过input标记设置的MAX\_FILE\_SIZE值。

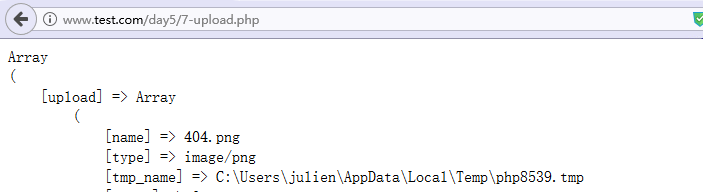
可以通过<input type="hidden" name="MAX\_FILE\_SIZE" value="30000" />实现浏览器对文件大小的初步判断，避免过大文件不必要的上传造成的浪费时间。

此隐藏域必须在文件域之前设置。



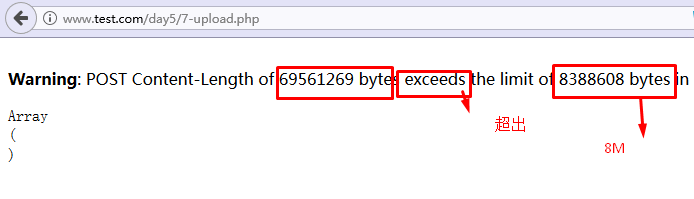


* 4，没有文件被上传
* 6，找不到临时文件夹



**说明：**

* 超出POST\_MAX\_SIZE时，数组$\_FILES为空，此时无错误代码提示。



超过UPLOAD\_MAX\_FILESIZE时，但是没有超过8M，$\_FILES中可以查看error信息。

## 文件上传的完整步骤

* 设计表单，重点注意文件域
* 接收数据：判断$\_FILES是否为空
* 判断错误：$\_FILES[‘myFile’][‘error’].出错提示错误信息
* 判断文件时是否是HTTP POST上传的文件：

is\_uploaded\_file(临时文件路径$\_FILES[‘myFile’][‘tmp\_name])；

//返回true（合法的POST文件） 或者false

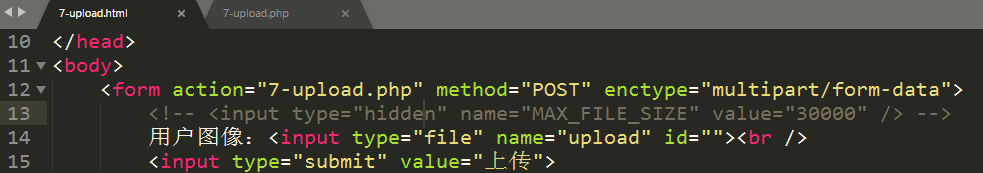
* 此时文件在临时路径，移动到网站永久路径

move\_uploaded\_file(临时路径$\_FILES[‘myFile’][‘tmp\_name]，永久路径)；

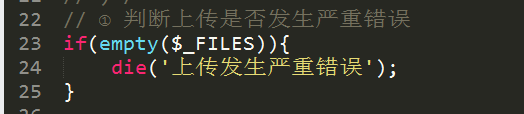
## 文件上传入门案例：

要求：出错则错误提示。成功，将文件存放在服务器中的upload目录

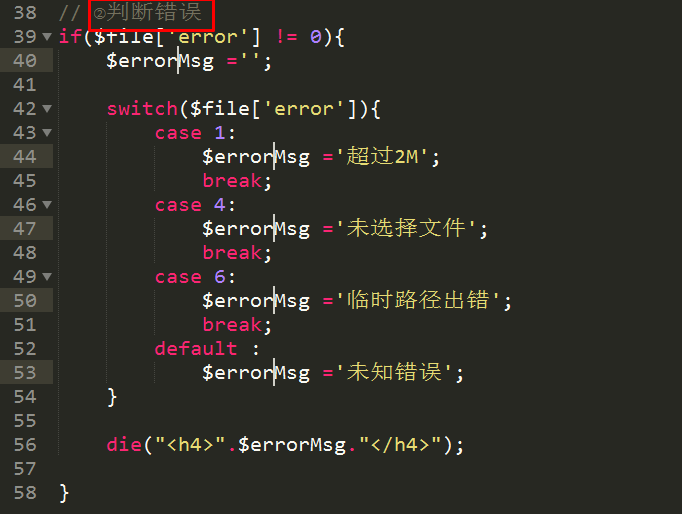
### 设计表单



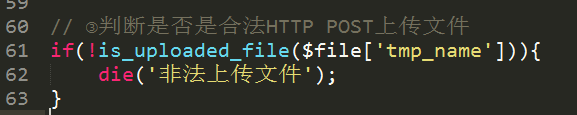
### 判断数组是否为空



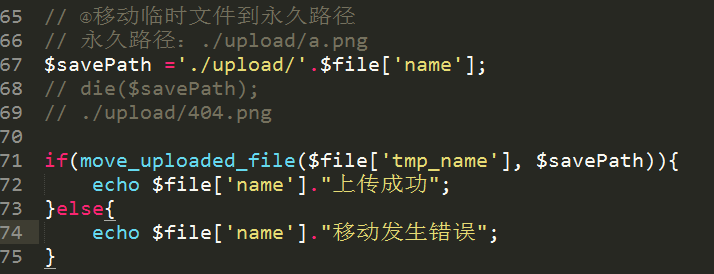
### 错误提示



### 判断是否是合法的HTTP post上传文件



### 移动至永久目录



# 文件上传的相关问题

## 文件扩展名的简单获取方法

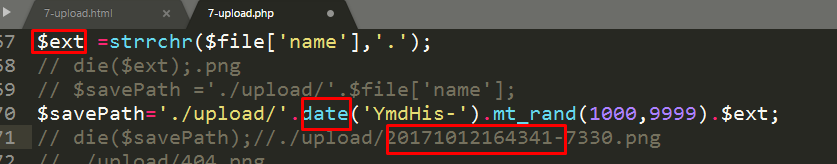
**截取.之后的字符串**：

string **strrchr** ( string $haystack , [mixed](mk:@MSITStore:E:\DocRef\WEB开发必备参考资料\php手册2015.chm::/res/language.pseudo-types.html#language.types.mixed) $needle )

在$haystack中查找$needle.找到后，返回之后的字符串。

## 避免文件名重复

对于同一网站的保存路径，不同的用户上传的文件可能重名。为了避免同名文件相互覆盖，需要给文件进行再次命名。可以使用当前日期时间及随机数生成



## 大小限定

临时文件单位为byte,比较时需要转换单位



## 文件类型限定（MIME类型）



网站支持的文件类型一般是固定的，需要判断上传文件的类型。

由于文件本身的扩展名可以修改，修改文件的扩展名之后，上传之后类型也发生改变。不能通过上传后的扩展名来判断文件的类型。只能通过Mime类型。

MIME： (Multipurpose Internet Mail Extensions) 是描述消息内容类型的因特网标准。MIME 消息能包含文本、图像、音频、视频以及其他应用程序专用的数据。不同的系统支持不同的MIME类型。

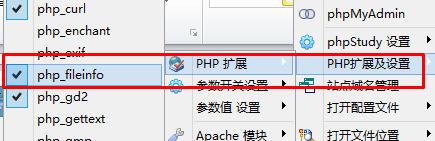
**PHP中常用的MIME类型：**

* image/png,image/gif,image/jpeg,
* text/html,text/css,text/javascript,
* text/plain，
* application/pdf

### 获取文件MIME类型步骤

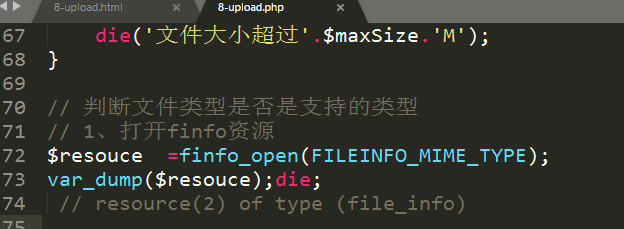
利用php\_fileinfo.dll（函数仓库）对MIME类型进行严格检查

①在PHP的扩展文件中开启php\_fileinfo.dll扩展



②打开fileinfo资源，为后续获取MIME类型准备

finfo\_open(FILEINFO\_MIME\_TYPE)；

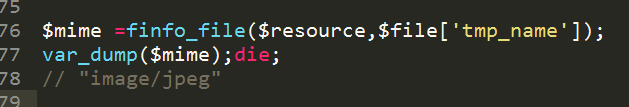


该函数执行后，执行成功返回资源型，失败返回false.

③读取上传文件的MIME类型

finfo\_file(finfo文件资源，临时文件名)；

返回文件的MIME类型字符串。



使用到的函数：

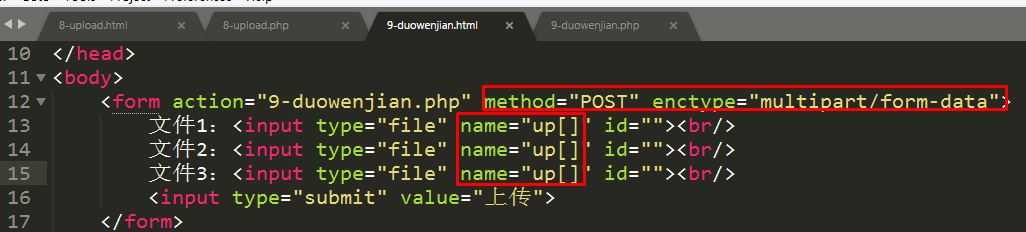
bool **in\_array** ( [mixed](mk:@MSITStore:E:\DocRef\WEB开发必备参考资料\php手册2015.chm::/res/language.pseudo-types.html#language.types.mixed) $needle , array $haystack )



# 多文件上传

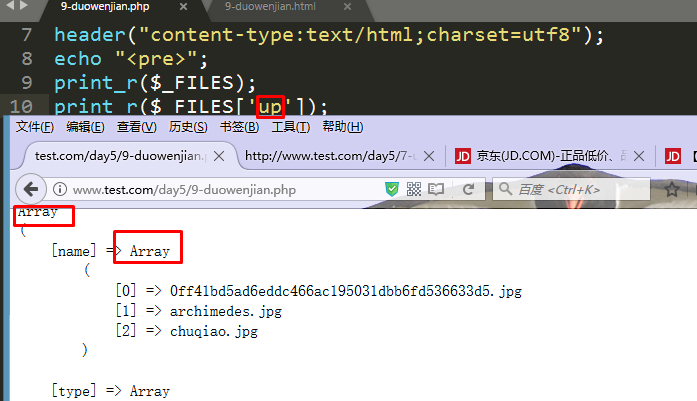
## 设计表单

文件域name值的设置使用数组格式，其他的同单文件上传的一样。



## 接收数据

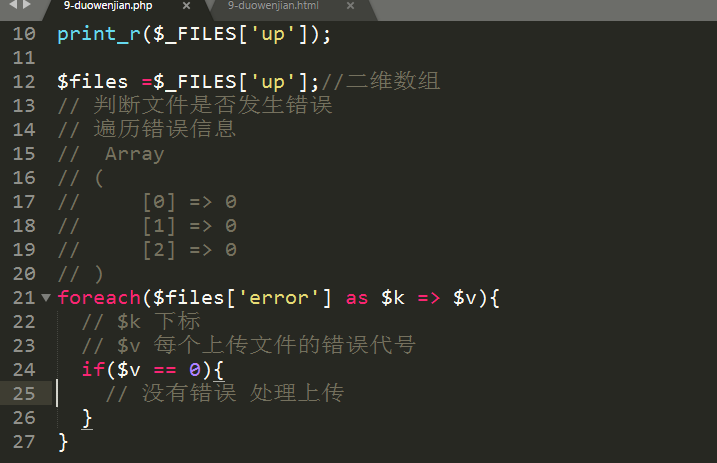
直接打印$\_FILES为三维数组。可以使用$\_FILES[‘myFile’]。系统自动将不同的文件的信息放在一起，进行单文件处理时，需要循环获取文件的完整信息。



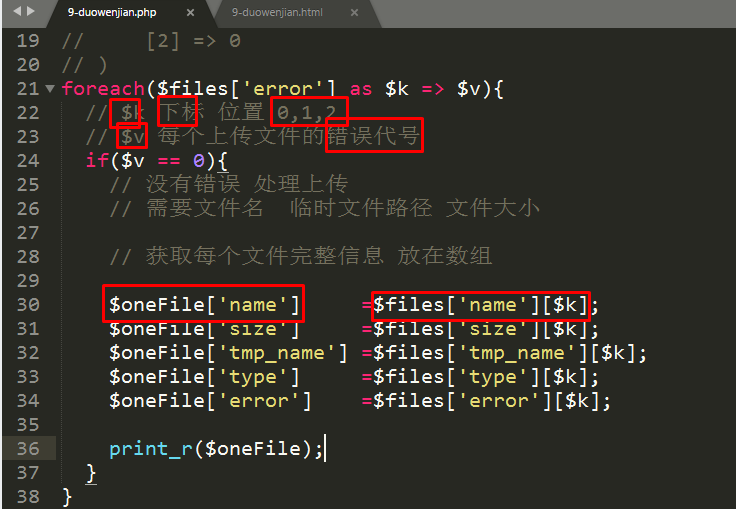
## 获取文件的完整信息



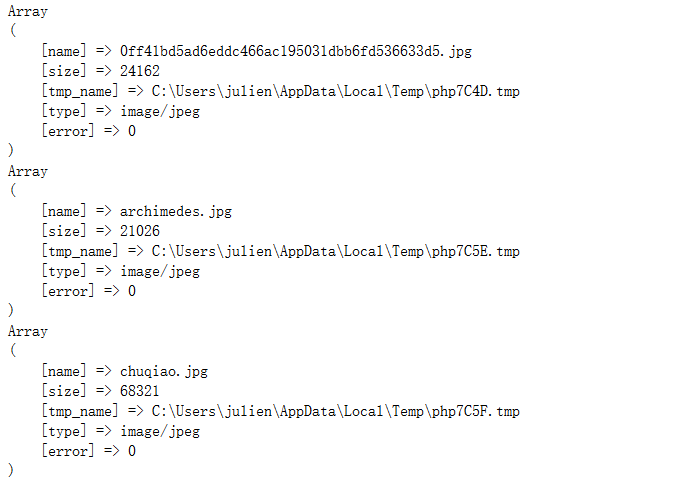
### 判断文件是否发生错误



### 获取每个文件的完整信息



**完整信息：**



### 处理单个文件上传

