**PHP核心编程第一天—表单传值**

目录

[一、 表单传值 3](#_Toc484625863)

[1. 应用场景 3](#_Toc484625864)

[2. 传值方式 3](#_Toc484625865)

[1) 分类 3](#_Toc484625866)

[二、 GET方式传值 3](#_Toc484625867)

[1. 表单方式 3](#_Toc484625868)

[2. 链接方式: 4](#_Toc484625869)

[1) html的a标记： 4](#_Toc484625870)

[2) 入门小案例：购物车初步 4](#_Toc484625871)

[3. JS跳转: 4](#_Toc484625872)

[1) location.href属性 4](#_Toc484625873)

[2) location.assign()方法 5](#_Toc484625874)

[4. PHP跳转: 5](#_Toc484625875)

[1) header的location选项 6](#_Toc484625876)

[2) header的refresh选项 6](#_Toc484625877)

[3) 封装跳转提示函数 6](#_Toc484625878)

[三、 POST方式 7](#_Toc484625879)

[1. 表单设计 7](#_Toc484625880)

[2. 接收数据 8](#_Toc484625881)

[3. POST的使用场景： 8](#_Toc484625882)

[四、 PHP接收数据的三种方式 8](#_Toc484625883)

[1. 预定义变量回顾 8](#_Toc484625884)

[1) 常见的预定义变量（重点） 8](#_Toc484625885)

[2) $\_SERVER的必知必会 9](#_Toc484625886)

[3) 封装数组打印工具函数 9](#_Toc484625887)

[2. $\_GET 10](#_Toc484625888)

[1) 注意事项 10](#_Toc484625889)

[2) 特例 10](#_Toc484625890)

[3. $\_POST 11](#_Toc484625891)

[1) 注意事项 11](#_Toc484625892)

[2) 案例：用户登录 11](#_Toc484625893)

[4. $\_REQUEST 12](#_Toc484625894)

[1) 注意事项 12](#_Toc484625895)

[2) 案例：同时接收GET及POST数据 12](#_Toc484625896)

[5. POST/GET方式的比较 13](#_Toc484625897)

[1) GET方式 13](#_Toc484625898)

[2) POST方式 13](#_Toc484625899)

[五、 PHP接收表单单选、复选框数据 13](#_Toc484625900)

[1. 下拉菜单的数据接收 13](#_Toc484625901)

[1) 表单域设置的注意事项 13](#_Toc484625902)

[2) 数据回显时的默认选中 14](#_Toc484625903)

[2. 单选项的数据接收 14](#_Toc484625904)

[1) 表单域设置的注意事项 14](#_Toc484625905)

[2) 数据回显时的默认选中 15](#_Toc484625906)

[3. 复选框的数据接收 15](#_Toc484625907)

[1) 表单注意事项 15](#_Toc484625908)

[2) 提交后的后续数据处理 15](#_Toc484625909)

[六、 应用：设计添加文章表单，并接收数据 15](#_Toc484625910)

[1) 设计添加文章数据表 15](#_Toc484625911)

[2) 设计表单 16](#_Toc484625912)

[3) ueditor插件的部署 17](#_Toc484625913)

[4) 实现数据的自动保存至变量 18](#_Toc484625914)

[七、 单文件上传 18](#_Toc484625915)

[1. 表单注意事项 18](#_Toc484625916)

[2. $\_FILES详解 18](#_Toc484625917)

[1) $\_FILES[‘myFile’][‘name’] 19](#_Toc484625918)

[2) $\_FILES[‘myFile’][‘tmp\_name’] 19](#_Toc484625919)

[3) $\_FILES[‘myFile’][‘type’] 19](#_Toc484625920)

[4) $\_FILES[‘myFile’]’[‘error’] 19](#_Toc484625921)

[5) $\_FILES[‘myFile’][‘size’] 19](#_Toc484625922)

[3. error错误代码含义 19](#_Toc484625923)

[4. 文件上传的完整步骤 20](#_Toc484625924)

[5. 文件上传入门案例： 21](#_Toc484625925)

[6. 文件上传的相关问题 21](#_Toc484625926)

[1) 文件扩展名的获取方法 21](#_Toc484625927)

[2) 避免文件名重复 22](#_Toc484625928)

[3) 大小限定 22](#_Toc484625929)

[4) 格式限定（MIME类型） 22](#_Toc484625930)

[5) 获取文件MIME类型步骤 23](#_Toc484625931)

[7. 应用：单文件上传函数封装 24](#_Toc484625932)

[八、 多文件上传 25](#_Toc484625933)

[1. 设计表单 25](#_Toc484625934)

[2. 接收数据 25](#_Toc484625935)

[3. 获取文件的完整信息 26](#_Toc484625936)

思考：

* GET方式提交数据的四种方式
* POST方式提交，表单的注意事项
* GET方式与POST提交数据的区别
* 接收表单数据的三个超全局变量
* 单选项、下拉菜单、复选项的表单设计
* 文件域的表单设计
* 单文件上传的数据处理
* 多文件上传的数据处理

# 表单传值

## 应用场景



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。

### 分类

GET方式、POST方式

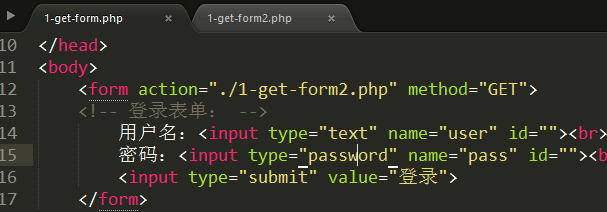
# GET方式传值

为表单的默认提交方式。

## 表单方式

form 表单的METHOD设置为GET或者不设置。name属性值必须填写。

登录表单：





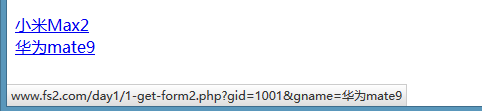
## 链接方式:

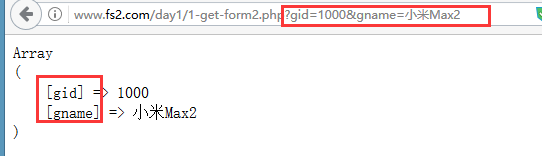
### html的a标记：

<a href=’请求的地址?a=100&b=200’>点击提交</a>

### 入门小案例：购物车初步







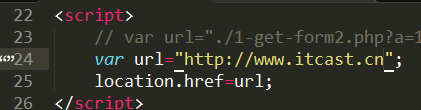
## JS跳转:

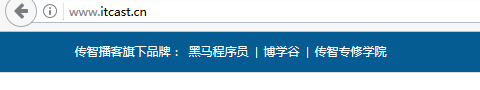
location为JS对象。对象有属性（特点）、方法（能做的事情，函数）

### location.href属性

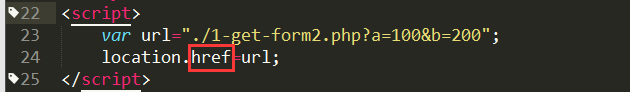
location.href =”请求的地址?a=100&b=200”

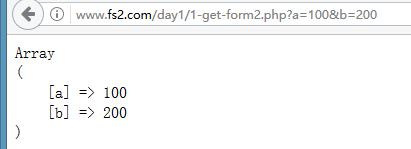
**站外跳转：**





**站内跳转：**

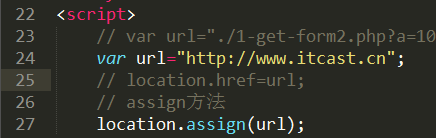


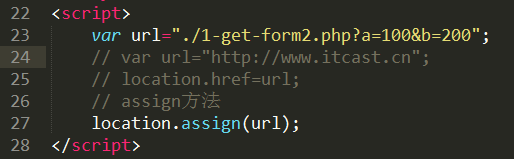


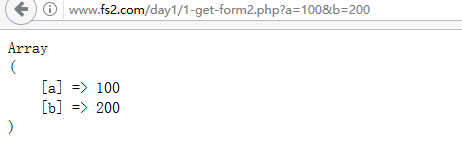
### location.assign()方法

方法的本质为函数，实现特定的功能。

location.assign(url);



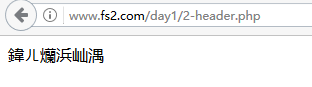




## PHP跳转:

header（）:服务器通知浏览器一些事项(字符集、如何跳转)

乱码：



**解决方案：**



### header的location选项

header(“location:路径”);

header后面的代码不执行。



强调：

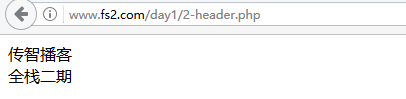
单引号不能解析变量。

### header的refresh选项

header(“refresh:3;url=地址”);

等待3秒后，跳转至对应地址。等待期间，继续执行后续代码。

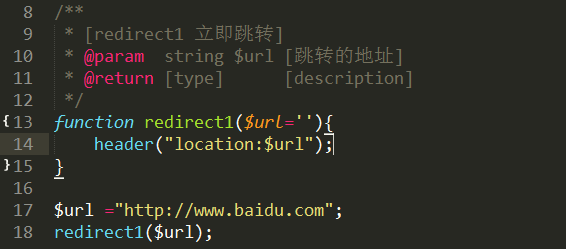






### 封装跳转提示函数

立即跳转及带弹框的跳转。functions.php





# POST方式

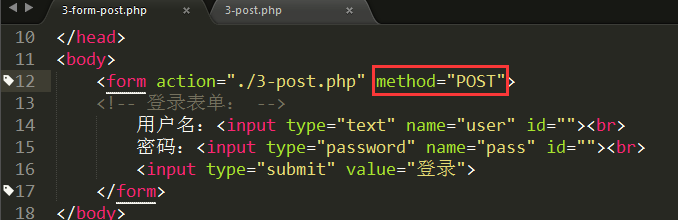


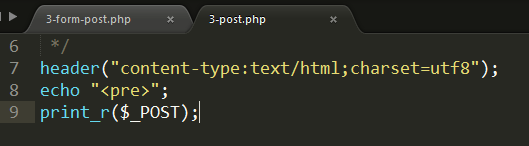
GET方式提交数据通过地址栏实现，数据安全性低。提交的数据量较小。对于密码等隐私信息、文件上传必须使用POST方式提交。

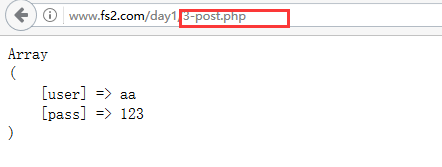
## 表单设计

表单的method为POST。保证name设置了属性值。

登录表单：







## 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。

## 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

$\_GET,$\_POST分别用来接收表单GET、POST数据。

## 预定义变量回顾

预定义变量：系统预先定义好的变量。

### 常见的预定义变量（重点）

变量名区分大小写。

预定义变量都是数组。

$\_SERVER:服务器环境信息

$\_GET:用户GET方式提交的数据。

$\_POST: 用户POST方式提交的数据。

$\_REQUEST:get及POST及COOKIE数据。

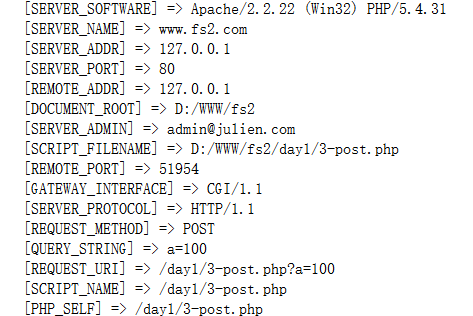
$\_COOKIE:保存COOKIE数据（保存在浏览器端的数据）。

$\_SESSION:保存SESSION数据（保存服务器端）。

$GLOBALS:引用超全局变量。

$\_FILES:上传的文件信息（文件名、文件类型、大小）

### $\_SERVER的必知必会



软件版本：$\_SERVER[‘SERVER\_SOFTWARE’];

服务器名称: $\_SERVER[‘SERVER\_NAME’];

服务器地址: $\_SERVER[‘SERVER\_ADDR’];

客户端地址: $\_SERVER[‘REMOTE\_ADDR’];

网站根目录: $\_SERVER[‘DOCUMENT\_ROOT’];

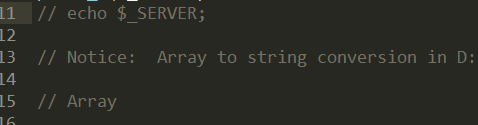
请求URI：$\_SERVER[‘REQUEST\_URI’];

请求的脚本名称：$\_SERVER[‘SCRIPT\_NAME’];

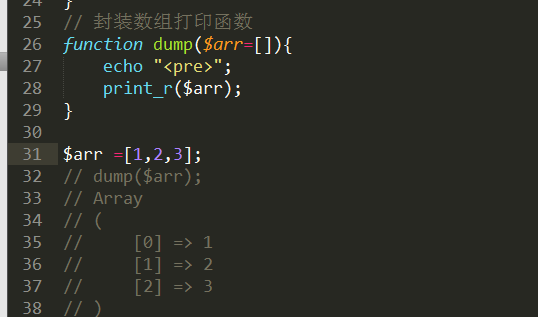
查询字符串:提交的参数。$\_SERVER[‘QUERY\_STRING’];

### 封装数组打印工具函数

数组不能直接使用echo语句输出。



可以通过var\_dump,print\_r。（研究var\_dump,print\_r区别）



其他页面使用该函数：直接使用include、require语句载入即可。

include:加载失败返回warning。后续代码继续执行。

require：加载失败，返回fatal error.后续代码不执行。

\_once:文件只加载一次。减少重复 代码，避免函数的重复定义。

墙裂推荐：



## $\_GET

### 注意事项

form表单：method=GET,name属性值必填。

a链接：

<a href=’./aa.php?a=100&b=200’>点击提交</a>

JS跳转：

location.href =地址;

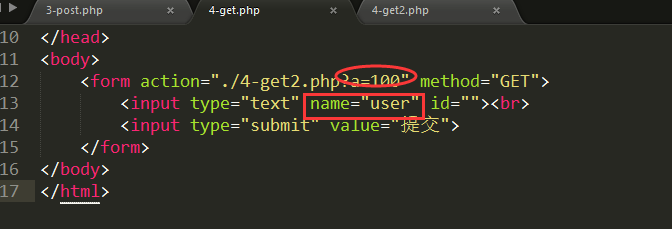
location.assign(地址);

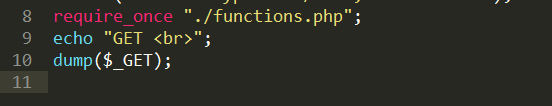
header函数:

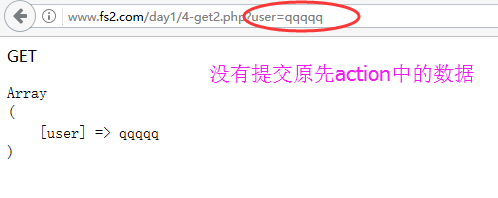
header(“location:地址”);

### 特例

表单使用GET方式提交数据时，无法接收同时通过action中提交的GET数据。







## $\_POST

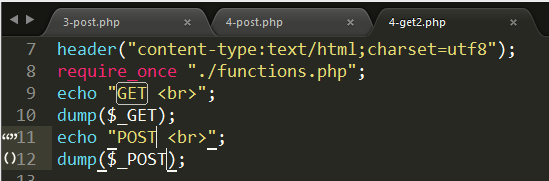
对表单数据安全性要求比较、数据量比较大。

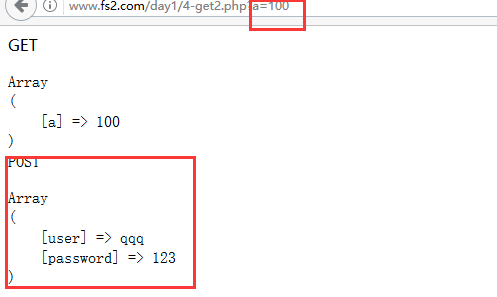
### 注意事项

form需要填写method=”POST”

### 案例：用户登录







## $\_REQUEST

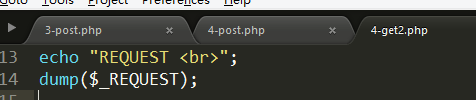
### 注意事项

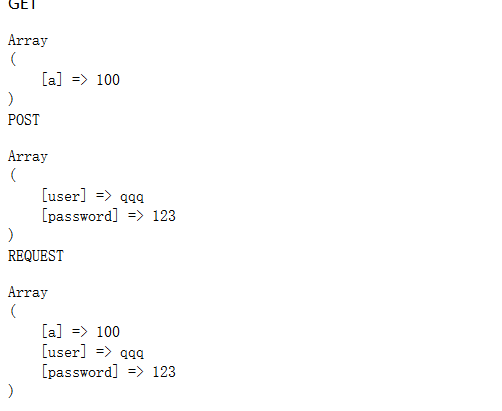
* $\_REQUEST =$\_GET + $\_POST +$\_COOKIE。
* Cookie中存储的一般为浏览器端保存的用户名及密码等信息。



* POST与GET提交的元素名称重复时，POST方式提交的数据覆盖GET数据。

### 案例：同时接收GET及POST数据





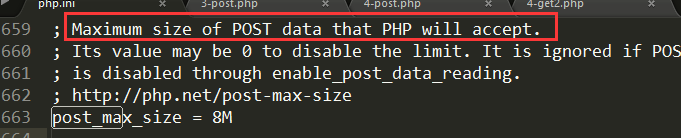
## POST/GET方式的比较

### GET方式

* 地址栏提交数据或者表单域(链接)提交（method=”GET”）。最终都是附在地址栏提交。安全性低。
* 数据上限一般为2kb,数据量小

### 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（post\_max\_size）



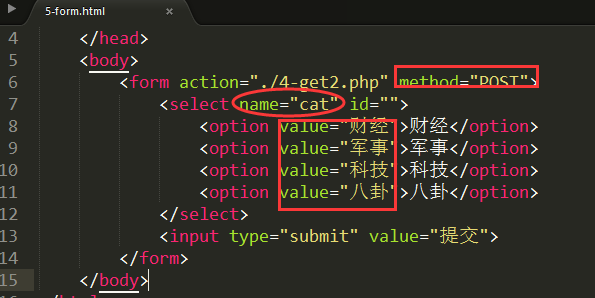
# PHP接收表单单选、复选框数据

表单域：文本域、单选按钮、多选按钮（复选项）、下拉菜单、多行文本

## 下拉菜单的数据接收

接收选项比较多的情况：籍贯、类型。

### 表单域设置的注意事项



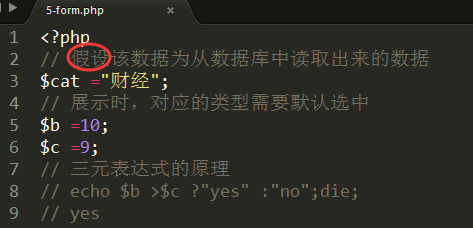
select option实现的是单选项，下拉菜单。

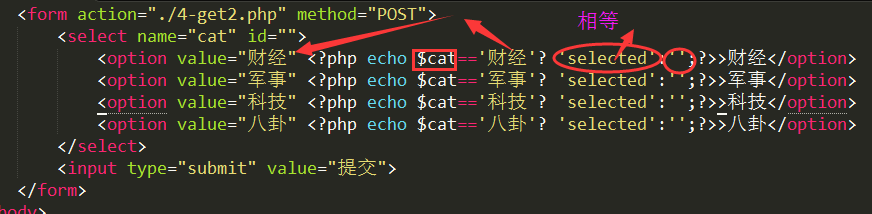
一般用于选项比较多的情况，如城市，省份，产品类型等等。

* name属性必选。
* value为必选。
* 默认选中：selected=’selected’



### 数据回显时的默认选中



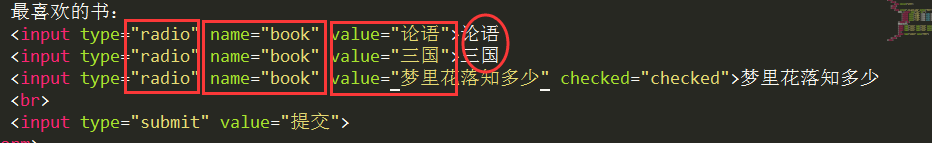


## 单选项的数据接收

### 表单域设置的注意事项

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

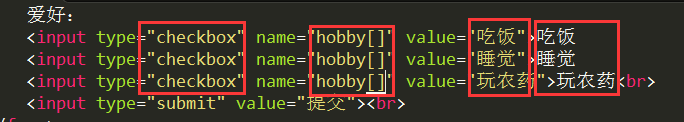


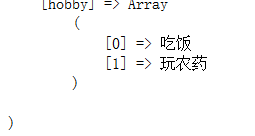
## 复选框的数据接收

type =checkbox,多选项。

### 表单注意事项

* name命名形式必须为：name[]，最终数据才能以数组的格式提交。不同的选项值，以数组元素的形式提交。
* value必选
* checked可选

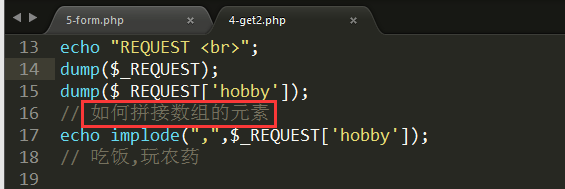




### 提交后的后续数据处理

使用implode函数拼接提交的多选项：

string **implode** ( string $glue , array $pieces )



# 应用：设计添加文章表单，并接收数据

表单：提供用户录入信息的窗口。

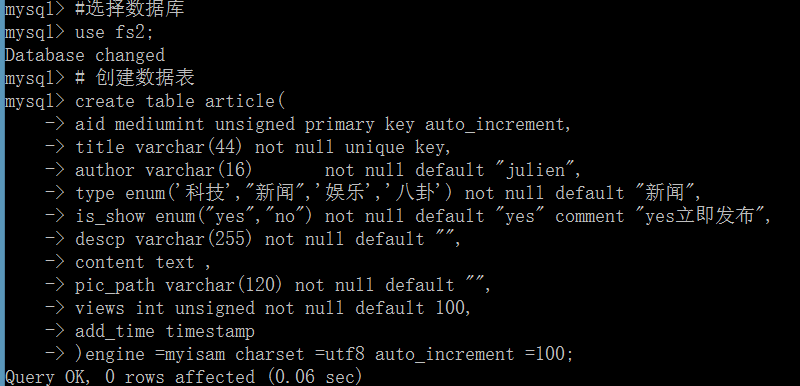
数据表：存储用户提交的数据。

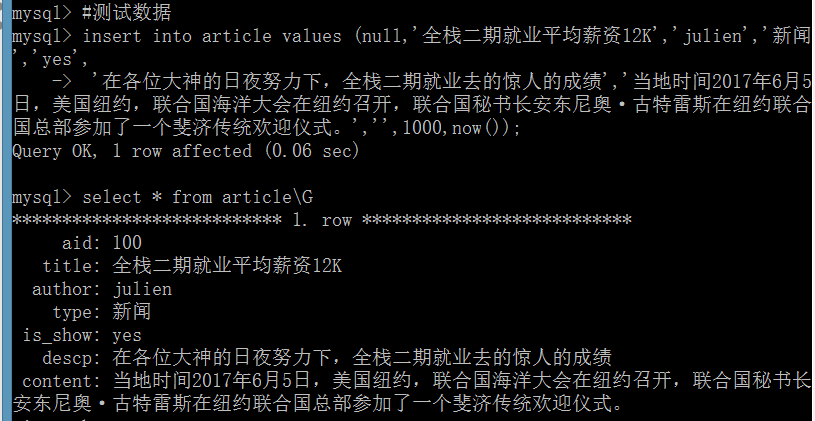
### 设计添加文章数据表

存储数据包括：标题，作者，栏目类型（科技，娱乐，新闻，八卦），是否立即发布（yes表示立即发布，no表示存为草稿），摘要，文章内容，图片路径，浏览量，发布时间

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **类型** | **属性** | **备注** |
| aid | mediumint unsigned | primary key auto\_increment | 文章编号 |
| title | varchar(44) | not null unique key | 标题 |
| author | varchar(16) | not null default “admin” | 作者 |
| type | enum(“科技”,”娱乐”,”新闻”,’八卦’) | not null default “新闻” | 类型 |
| is\_show | enum(‘yes’,’no’) | not null default “yes” | 是否发布：yes立即发布，no存为草稿 |
| descp | varchar(255) | not null default “” | 摘要 |
| content | text |  | 内容 |
| pic\_path | varchar(120) | not null default “” | 图片路径 |
| views | int unsigned | not null default 100 | 点击量 |
| add\_time | timestamp |  | 发布时间 |

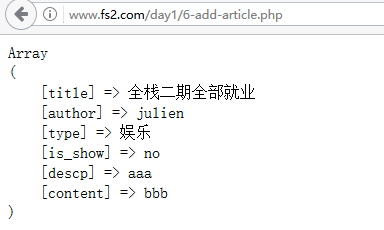
**创建数据库及数据表：**





### 设计表单





### ueditor插件的部署

UEditor是由百度web前端研发部开发所见即所得富文本web编辑器，具有轻量，可定制，注重用户体验等特点，开源基于MIT协议，允许自由使用和修改代码。

引入的代码：

<!-- 富文本编辑器ueditor -->

<script charset="utf-8" src="ueditor/kindeditor-min.js"></script>

<script charset="utf-8" src="ueditor/lang/zh\_CN.js"></script>

<script>

var editor;

KindEditor.ready(function(K) {

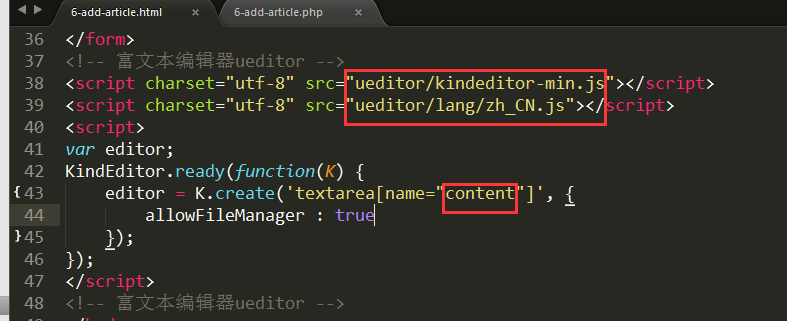
editor = K.create('textarea[name="cont"]', {

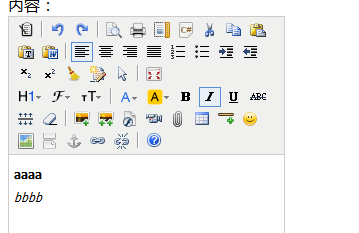
allowFileManager : true

});

});

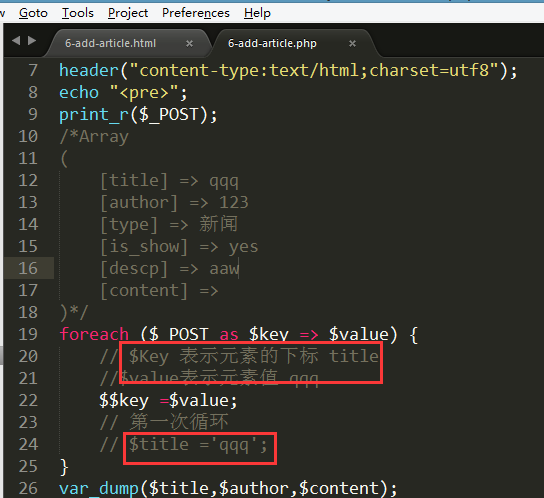
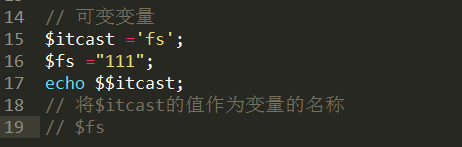
</script>





### 实现数据的自动保存至变量

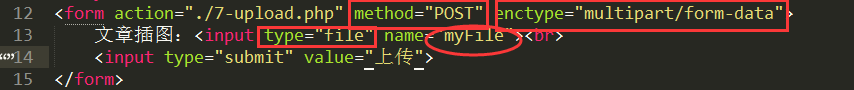
**可变变量的原理：**



# 单文件上传

应用场景：商城或者新闻网展示相关的图片。

## 表单注意事项

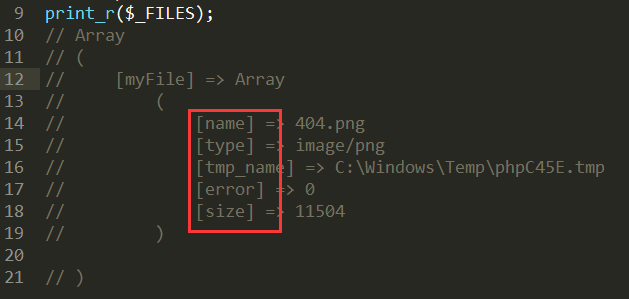


* form 的 数据的编码的类型enctype 等于multipart/form-data。

enctype 属性，默认以 application/x-www-form-urlencoded方式编码。

* 数据提交方式：POST方式
* 文件域：<input type=”file” name=”属性值”/>

## $\_FILES详解



使用预定义变量$\_FILES接收文件上传信息。$\_FILES为超全局数组。假设myFile为文件域的name属性值。

### $\_FILES[‘myFile’][‘name’]

文件的原始名称。

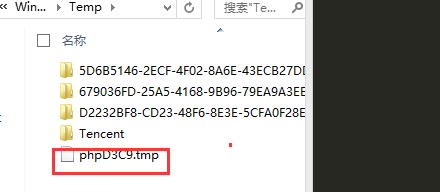
### $\_FILES[‘myFile’][‘tmp\_name’]

文件的临时路径，默认为c:/windows/temp（运行窗口—>temp）。没有直接将文件上传至Apache服务器的根目录：用户的数据是不安全。

临时文件保存路径可以在PHP.ini中upload\_tmp\_dir选项进行设置。未指定时使用系统的默认存储路径c:/windows/temp进行存储。

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：

image/jpeg,image/png,image/gif,text/html

### $\_FILES[‘myFile’]’[‘error’]

文件上传的错误代码。0代表没有发生错误。

### $\_FILES[‘myFile’][‘size’]

文件大小（单位为byte，字节）

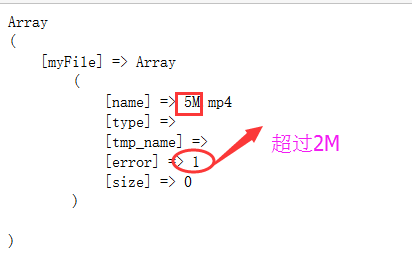
1M =1024 KB

1kb =1024byte

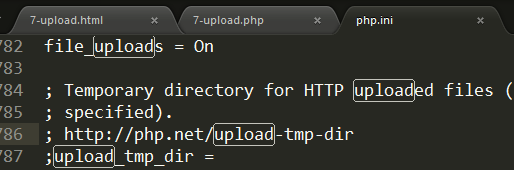
1byte =8bit位

## error错误代码含义

* 0：文件上传至临时路径成功
* 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（超过2M,不超过8M）

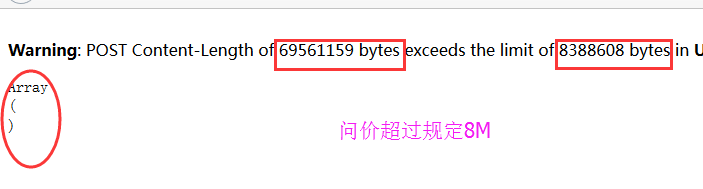


* 4，没有文件被上传
* 6，找不到临时文件夹

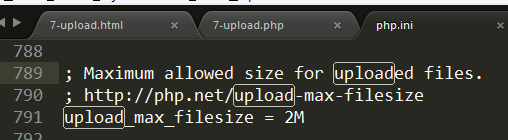


**说明：**

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
* 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：

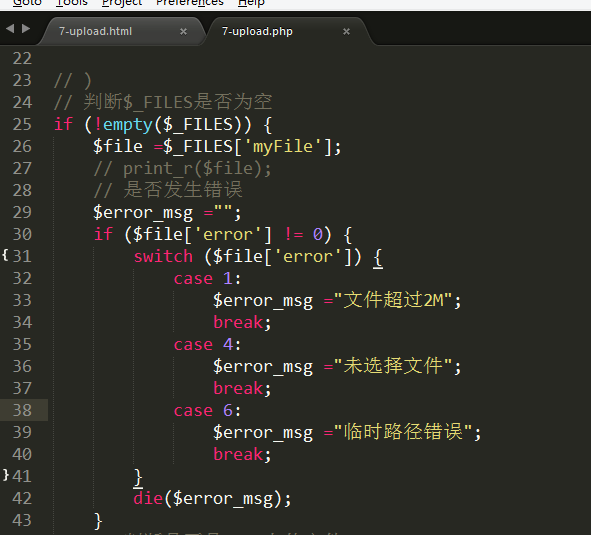
is\_uploaded\_file(临时文件路径)；

* 此时文件在临时路径，移动到网站永久路径

move\_upload\_file(临时路径，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





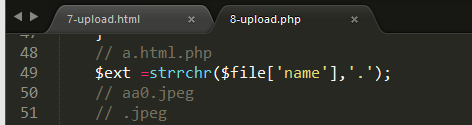
## 文件上传的相关问题

### 文件扩展名的获取方法

**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 )

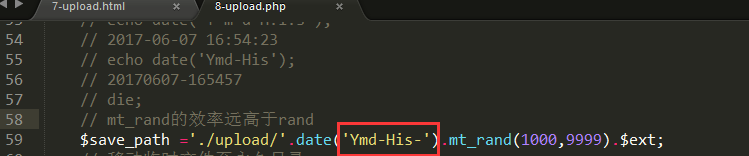
在$haystack中查找$needle.找到后，返回之后的字符串。

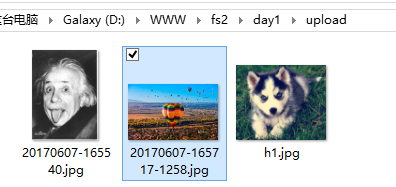


### 避免文件名重复

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，所以需要给文件进行命名。

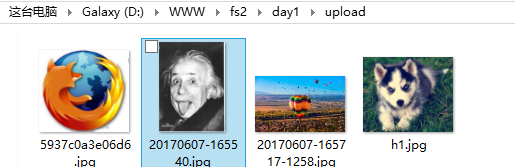
* 使用当前时间命名及添加随机数。





* 使用uniqid函数



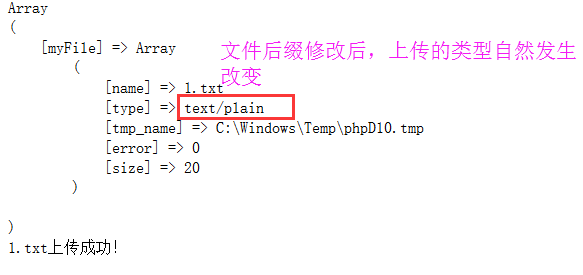


### 大小限定

临时文件单位为byte,比较时需要转换单位



### 格式限定（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。不可以使用name值判断类型。

* 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。
* 常见的MIME类型：

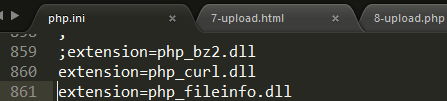
image/png,image/gif,image/jpeg,text/html,text/css,text/javascript,text/plain

### 获取文件MIME类型步骤

* 利用php\_fileinfo.dll对MIME类型进行严格检查

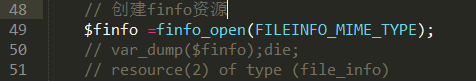
dll文件：动态链路库（dynamic link library）。包含大量函数的函数库。可以按照需求调用。

①在php.ini开启php\_fileinfo.dll扩展



②打开fileinfo资源

finfo\_open(FILEINFO\_MIME\_TYPE)；



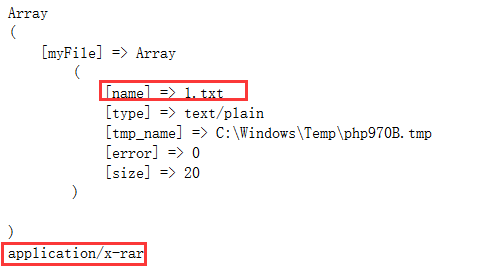
成功返回资源型，失败返回false.

资源是到第三方数据的引用，数据本身不PHP的（建立成功的数据库连接，返回为资源类型，相当于到数据库数据的一个引用）。

③获取上传文件的MIME类型

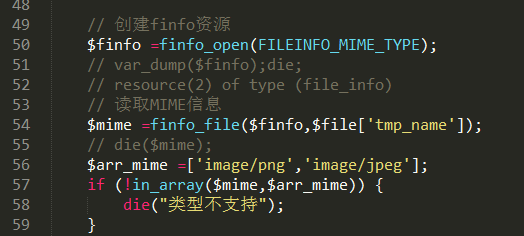
finfo\_file(文件资源，临时文件名)；

返回文件的MIME类型字符串。



**在数组中查找元素：**

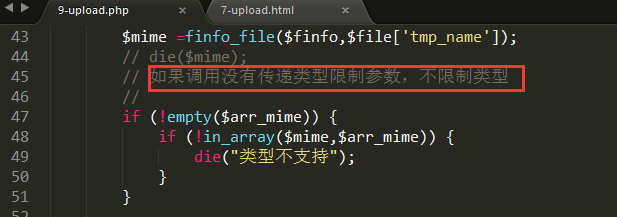
bool **in\_array** ( [mixed](mk:@MSITStore:E:\DocRef\WEB开发必备参考资料\php手册2015.chm::/res/language.pseudo-types.html#language.types.mixed) $needle , array $haystack [, bool $strict = **FALSE** ] )



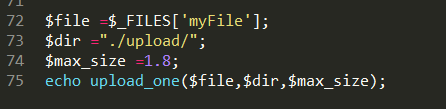
## 应用：单文件上传函数封装

函数定义要求：实现文件类型，大小的判断，实现路径自定义。





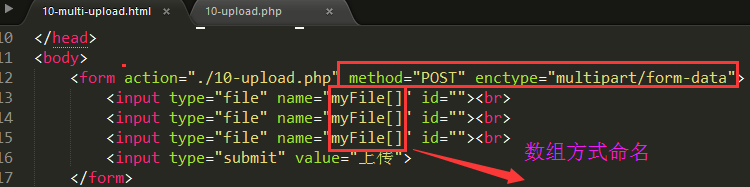




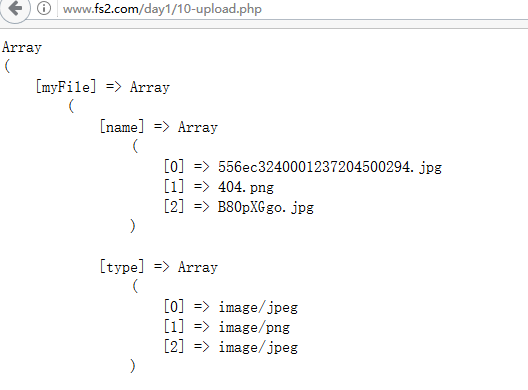
# 多文件上传

## 设计表单

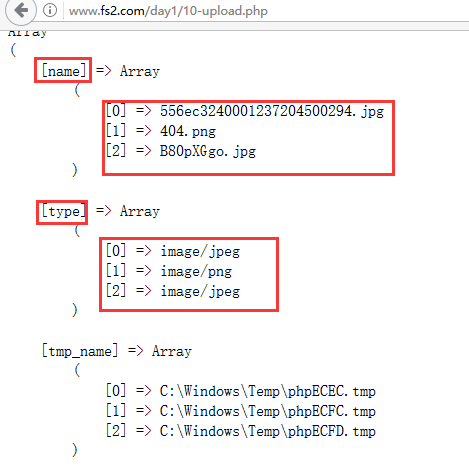
name值的设置使用数组格式。



## 接收数据



直接打印$\_FILES为三维数组。可以使用$\_FILES[‘myFile’]。



## 获取文件的完整信息

