

文件上传

1、单文件上传

需求分析:

在计算机中，添加用户账号时，为了使用户的形象更加的具体、鲜活，经常需要设置头像。同样，在Web开发过程中，也经常需为某个用户上传头像。

本案例将带领大家开发一个网站用户头像上传的功能，从而掌握PHP对上传文件的接收与处理等相关知识。

设计思路:

- (1) 编写HTML页面，在页面中创建一个表单，用于上传用户头像。
- (2) 使用数组保存用户基本信息，并将其显示在头像上传表单中。
- (3) 在浏览器中访问用户注册页面，选择上传的文件后提交表单。
- (4) 通过PHP接收、处理上传文件信息，并同时显示上传的头像。



1、文件上传表单

(1) enctype属性设置

在表单中，要想实现文件上传，需要将enctype属性的值设置为“multipart/form-data”，让浏览器知道在表单信息中除了其他数据外，还有上传的文件数据。

(2) 表单提交方式设置

理由：浏览器会将表单提交的数据（除了文件数据外）进行字符编码，并单独对上传的文件进行二进制编码，又因为在URL地址栏上不能传输二进制编码数据。

结论：要想实现文件上传表单，必须将表单提交方式设为POST方式。

示例：

```
<form action="fileupload.php" method="post" enctype="multipart/form-data">
    上传头像: <input type="file" name="upload" id="">
    <input type="submit" value="保存头像">
</form>
```

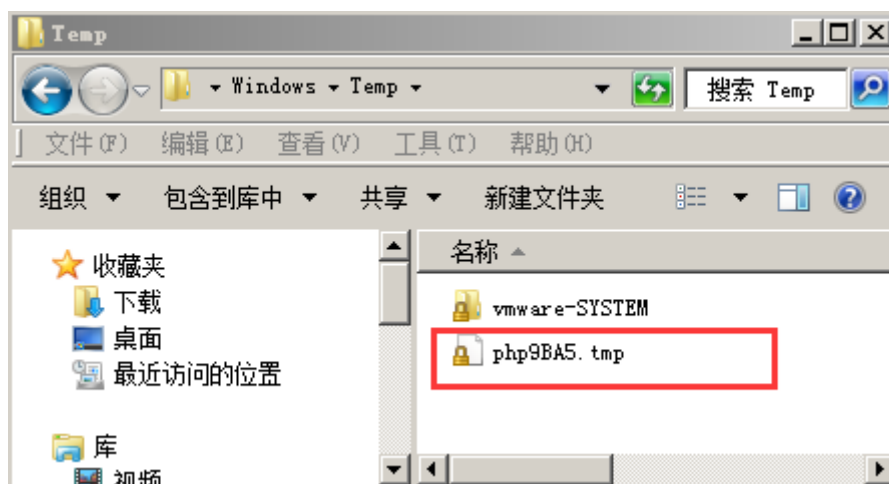
要想在服务器端获取上传文件信息，还需要为上传文件设置name属性。

2、处理上传文件

PHP默认将表单上传的文件保存到服务器系统的临时目录下，该临时文件的保存期为脚本周期。所谓脚本周期就是执行PHP文件所需的时间。

查看临时文件：

在处理表单的文件中，通过sleep(seconds)函数延迟PHP文件执行的时间，在“C:\Windows\Temp”目录中查看临时文件。



当PHP执行完毕后，图中方框内的临时文件就会被释放掉。

处理上传文件的步骤：

- (1) 利用PHP提供的超全局数组\$_FILES获取上传的临时文件信息。
- (2) 使用move_uploaded_file(临时文件，目标文件地址)函数，将临时文件保存到我们为其指定的目标文件地址中。

3、获取上传文件信息

在PHP获取上传文件时，使用\$_FILES二维数组来存储上传文件的信息，该数组的一维保存的是上传文件的名称，二维保存的是该上传文件的具体信息。

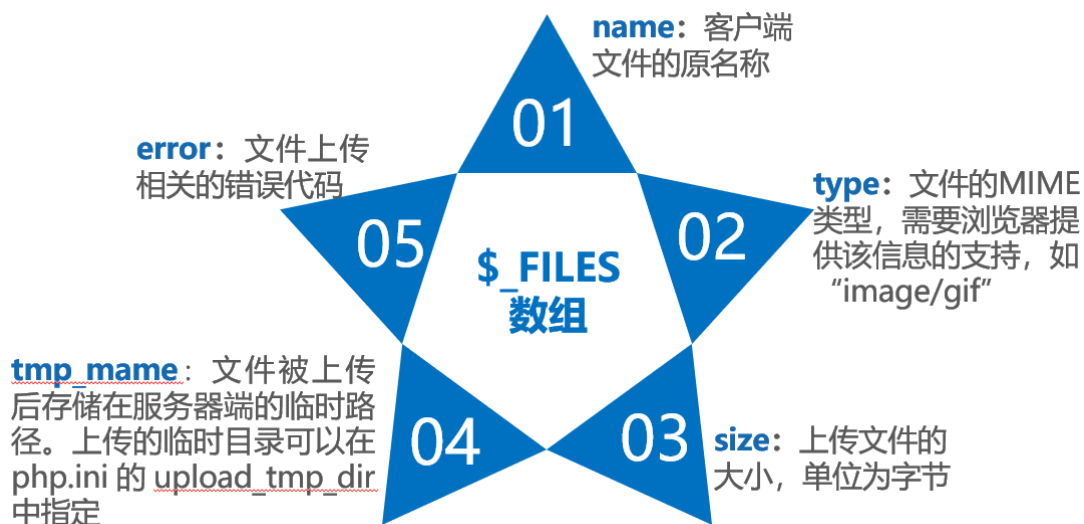
```
<?php
$upload=$_FILES['upload'];
var_dump($upload);
```

或者：

```

echo $_FILES['upload']['name'];
//输出上传文件大小: 879394
echo $_FILES['upload']['size'];
//输出上传文件结果: 0 (表示成功)
echo $_FILES['upload']['error'];
//输出上传文件类型: image/jpeg
echo $_FILES['upload']['type'];
//输出上传后临时文件名: C:\windows\Temp\phpCF84.tmp
echo $_FILES['upload']['tmp_name'];

```



文件上传错误代码:

代码	常量	说明
0	UPLOAD_ERR_OK	没有错误发生, 文件上传成功
1	UPLOAD_ERR_INI_SIZE	上传的文件超过了php.ini中upload_max_filesize 选项限制的值
2	UPLOAD_ERR_FORM_SIZE	上传文件的大小超过了HTML表单中MAX_FILE_SIZE选项指定的值
3	UPLOAD_ERR_PARTIAL	只有部分文件被上传
4	UPLOAD_ERR_NO_FILE	没有文件被上传
6	UPLOAD_ERR_NO_TMP_DIR	找不到临时目录
7	UPLOAD_ERR_CANT_WRITE	文件写入失败

4、判断上传文件类型

在实际开发中, 经常需要对用户上传的文件类型进行判断, 例如: 在电子商务网站中, 只允许用户上传jpg、png和gif格式的商品图片, 那么在处理上传文件时, 就需要对上传文件的类型进行判断。

```

//获取上传文件的类型
$type = $_FILES['upload']['type'];
//允许上传文件的类型
$allow_type = array('image/jpeg','image/png','image/gif');
//判断上传文件类型
if(!in_array($type,$allow_type)){
    echo '图像类型不符合要求，允许的类型为:'.implode(",",$allow_type);
    return false;
}

```

数组\$allow_type中的元素是MIME类型。

MIME类型指的是Internet内容类型描述的事实标准，“/”前面的部分表示数据的大类别，例如：图像image，声音audio等，后面的部分表示大类别下的具体类型。

5、用户头像上传代码：

1、fileupload.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        img{max-width: 400px;}
    </style>
</head>

<body>
    <div>
        <h2>编辑用户头像</h2>
        <p>用户姓名:
            <?php echo $info['name'];?>
        </p>
        <p>现有头像: </p>
        
        <form action="fileupload.php" method="post" enctype="multipart/form-
data">
            上传头像: <input type="file" name="upload" id="">
            <input type="submit" value="保存头像">
        </form>
    </div>
</body>

</html>

```

2、fileupload.php

```

<?php
header('Content-type:text/html;charset=utf-8');
$info = array('id' => 234, 'name' => '王五');
// var_dump($_FILES['upload']);

```

```

if (!empty($_FILES['upload'])) {
    $pic_info = $_FILES['upload'];
    //判断文件是否有错误
    if ($pic_info['error'] > 0) {
        $error_msg = '上传错误';
        switch ($pic_info['error']) {
            case 1:
                $error_msg .= '文件大小超过了php.ini中upload_max_file_size选项限制的
值';
                break;
            case 2:
                $error_msg .= '文件大小超过了表单中 max_filesize选项限制的值';
                break;
            case 3:
                $error_msg .= '文件只有部分被上传';
                break;
            case 4:
                $error_msg .= '没有文件被上传';
                break;
            case 6:
                $error_msg .= '找不到临时文件夹';
                break;
            case 7:
                $error_msg .= '文件写入失败';
                break;
            default:
                $error_msg .= '未知错误';
                break;
        }
        echo $error_msg;
        return false;
    }
    //获取文件类型
    $type=$pic_info['type'];
    $allow_type = array('image/jpeg', 'image/png', 'image/gif');
    //判断上传文件类型
    if (!in_array($type, $allow_type)) {
        echo '图像类型不符合要求，允许的类型为:' . implode(",", $allow_type);
        return false;
    }
    //允许上传文件的大小,1048576表示1M
    $max_size=1048576;
    $size=$pic_info['size'];
    if($size>$max_size){
        echo '文件不能超过'.$max_size.'字节';
        return false;
    }
    $new_file = $info['id'] . strrchr($pic_info['name'], '.');
    $filename = '../upload/' . $new_file;
    echo $filename;
    //若用户现有头像存在则删除
    if(file_exists($filename)){
        unlink($filename);
    }
    //头像上传到临时目录成功，将其保存到脚本所在目录下upload文件夹中
    if (!move_uploaded_file($pic_info['tmp_name'], $filename)) {
        echo '头像上传失败';
    }
}

```

```
}

require 'fileupload.html';
```

2、多文件上传

掌握多文件的上传，能够编写代码完成多文件的上传

- 1、一个页面有多处文件上传。上传的文件没有直接关联，本质还是单文件上传。
- 2、一个页面只有一个上传按钮。一次可以上传多个文件，上传的文件属于同一类。

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>文件上传</title>
</head>

<body>

    <form action="upload.php" method="post" enctype="multipart/form-data">
        <input type="hidden" name="MAX_FILE_SIZE" value="1000000">
        选择文件1: <input type="file" name="myfile[]"><br>
        选择文件2: <input type="file" name="myfile[]"><br>
        选择文件3: <input type="file" name="myfile[]"><br>
        <input type="submit" value="上传文件">
    </form>

</body>

</html>
```

处理多文件上传：使用\$_FILES接收上传文件的信息，利用循环处理文件信息。

```
<?php
    if (!empty($_POST)) {
        $allowtype = array("gif", "png", "jpg"); //允许上传的文件格式
        $size = 1000000; //设置允许大小1M以内的文件
        $path = "./upload"; //设置上传后保存文件的路径

        // echo '<pre>';
        // var_dump($_FILES['myfile']);
        //判断文件是否可以上传到服务器 $_FILES['myfile']['error']为0表示成功
        for ($i = 0; $i < count($_FILES['myfile']['name']); $i++) {
            $upfile[$i] = $_FILES['myfile']['name'][$i];
            if ($_FILES['myfile']['error'][$i] > 0) {
                echo "上传错误";
                switch ($_FILES['myfile']['error'][$i]) {

                    case 1:
                        die('第' . ($i + 1) . '个文件上传文件大小超出了PHP配置文件中的
                        约定值: upload_max_filesize');
                    case 2:
                        die('上传第' . ($i + 1) . '个文件大小超出了表单中的约定值:
                        MAX_FILE_SIZE');
```

```

        case 3:
            die('第' . ($i + 1) . '个文件只被部分上传');
        case 4:
            die('第' . ($i + 1) . '个文件没有上传');
        default:
            die('未知错误');
    }
}

//判断上传的文件是否为允许的文件类型，通过文件的后缀名
$hz[$i]=substr(strrchr($_FILES['myfile']['name'][$i], '.'),1);
if (!in_array($hz[$i], $allowtype)) {
    die("第" . ($i + 1) . "个文件后缀是<b>{$hz[$i]}</b>,不是允许的文件类型!");
}

//判断上传的文件是否允许大小
if ($_FILES['myfile']['size'][$i] > $size) {
    die("第" . ($i + 1) . "个文件超过了允许的<b>{$size}</b>");
}

//为了系统安全，同时也为了同名文件不被覆盖，上传后将文件名使用系统定义
$filename[$i] = date("YmdHis") . rand(100, 999) . "." . $hz[$i];
//判断是否为上传文件
if (is_uploaded_file($_FILES['myfile']['tmp_name'][$i])) {
    if (!move_uploaded_file($_FILES['myfile']['tmp_name'][$i], $path . '/' . $filename[$i])) {
        die("不能将文件移动到指定目录");
    }
} else {

    die("上传文件{$_FILES['myfile']['name'][$i]}不是一个合法文件");
}

//如果文件上传成功
$filesize[$i] = $_FILES['myfile']['size'][$i] / 1024;
echo "文件{$supfile[$i]}上传成功，保存在目录{$path}中，文件大小为{$filesize[$i]}KB<br>";
}
}

require 'upload.html';

```