核心编程第4天

[学习目标 2](#_Toc521514042)

[图像处理概述 2](#_Toc521514043)

[1、开启GD2图像扩展库 2](#_Toc521514044)

[2、查看图像扩展库GD2是否开启 2](#_Toc521514045)

[3、创建图像的大致步骤 3](#_Toc521514046)

[4、画布坐标系说明 3](#_Toc521514047)

[创建图像和销毁图像 3](#_Toc521514048)

[1、创建基于已有图像的图像imagecreatefromjpeg() 3](#_Toc521514049)

[2、创建空画布图像imagecreatetruecolor() 4](#_Toc521514050)

[3、销毁图像资源imagedestroy() 4](#_Toc521514051)

[图像操作 5](#_Toc521514052)

[1、为图像分配颜色imagecolorallocate() 5](#_Toc521514053)

[2、输出图像到浏览器或保存文件imagejpeg() 6](#_Toc521514054)

[3、水平地画一行字符串imagestring() 7](#_Toc521514055)

[4、获取画布的宽度和高度 8](#_Toc521514056)

[5、获取内置字体的宽度和高度 8](#_Toc521514057)

[6、实例：在图像上绘制一行居中的字符串 9](#_Toc521514058)

[7、画一矩形并填充 9](#_Toc521514059)

[8、画一个单一像素 10](#_Toc521514060)

[实例：图像验证码 10](#_Toc521514061)

[1、绘制图像验证码 10](#_Toc521514062)

[2、产生一个指定范围的数组range() 11](#_Toc521514063)

[3、合并数组array\_merge() 11](#_Toc521514064)

[4、从数组中随机取出一个或多个单元(下标) 11](#_Toc521514065)

[5、生成更好的随机数 12](#_Toc521514066)

[实例：往图像上写入一行汉字 12](#_Toc521514067)

[实例：制作图像水印效果 12](#_Toc521514068)

[实例：生成图像缩略图 13](#_Toc521514069)

[文件上传 14](#_Toc521514070)

[1、文件上传原理 14](#_Toc521514071)

[2、超全局变量数组 $\_FILES 15](#_Toc521514072)

[3、上传文件错误代码 15](#_Toc521514073)

[4、查看上传的临时文件位置 15](#_Toc521514074)

[5、将上传文件移动到新位置 16](#_Toc521514075)

[6、上传文件的相关配置(php.ini) 16](#_Toc521514076)

[7、获取文件路径信息 pathinfo() 17](#_Toc521514077)

[8、检查数组中是否存在某个值 in\_array() 17](#_Toc521514078)

[9、生成唯一ID uniqid() 17](#_Toc521514079)

[实例：单个文件上传 18](#_Toc521514080)

[1、上传表单制作upload.html 18](#_Toc521514081)

[2、上传文件的程序处理upload.php 18](#_Toc521514082)

[实例：多个文件上传 19](#_Toc521514083)

[1、制作上传表单 19](#_Toc521514084)

[2、上传多个文件的程序处理 20](#_Toc521514085)

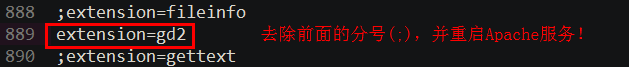
# 学习目标

* 图像处理概述
* 创建图像和销毁图像
* 实例：图像验证码
* 实例：图像水印效果
* 实例：图像缩略图效果
* 文件上传

# 图像处理概述

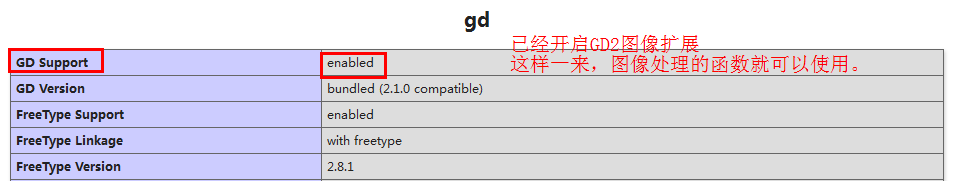
## 1、开启GD2图像扩展库

* PHP不仅限于只产生HTML的输出，还可以创建与操作多种不同格式的图像文件。PHP提供了一些内置的图像处理函数，也可以使用GD函数库创建新图像或处理已有的图像。目前GD2库支持JPEG、PNG和WBMP格式。
* GD扩展用于动态创建图片，使用C语言编写，开放源代码，现在的版本是2.0，所以称为GD2。
* 开启GD2扩展库：将php.ini中**extension=php\_gd2.dll**选项前的分号去掉，重启。



## 2、查看图像扩展库GD2是否开启





## 3、创建图像的大致步骤

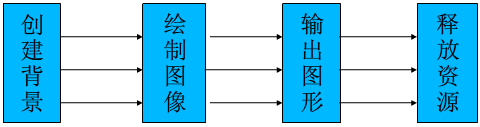
在PHP中创建一个图像，大致需要四个步骤：

（1）创建画布：创建一个画布，以后的操作都基于此画布操作；

（2）绘制图形：在画布上绘制图像轮廓或输入文本；

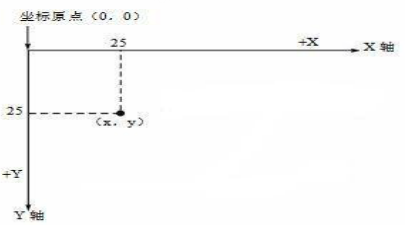
（3）输出图像：也可以另存为；

（4）释放资源：释放图像占用的内存资源。



## 4、画布坐标系说明

下图说明了画布的坐标系。坐标原点位于画布左上角，以像素为单位。



# 创建图像和销毁图像

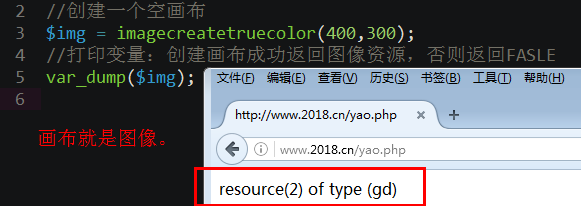
## 1、创建基于已有图像的图像imagecreatefromjpeg()

* 描述：由文件或 URL 创建一个新图象。
* 语法：**resource imagecreatefrompng ( string $filename )**
* 参数：$filename为图像的完整路径。
* 返回：成功后返回图象资源,失败后返回 FALSE 。
* 提示：imagecreatefromjpeg()和imagecreatefromgif()语法与该函数一样。



## 2、创建空画布图像imagecreatetruecolor()

* 描述：新建一个真彩色图像，支持24位色，即RGB(256,256,256)。
* 语法：**resource imagecreatetruecolor ( int $width , int $height )**
* 参数：$width图像宽度；$height图像高度；
* 返回：成功后返回图象资源,失败后返回 FALSE 。



## 3、销毁图像资源imagedestroy()

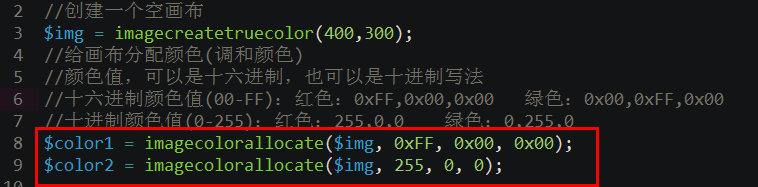
* 描述：销毁一图像。释放与 image图像标识符关联的内存。
* 语法：**bool imagedestroy ( resource $image )**
* 参数：$image为由imagecreatetruecolor()创建的图像标识符。

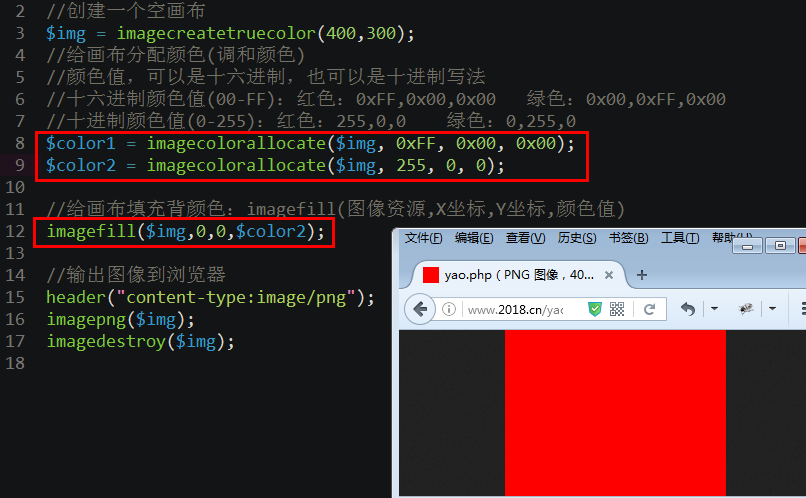


# 图像操作

## 1、为图像分配颜色imagecolorallocate()

* 语法：**int imagecolorallocate(resource $image,int $red,int $green,int $blue)**
* 参数：$image图像资源标识符；
* 提示：第一次对 imagecolorallocate()的调用会给图像填充背景色。

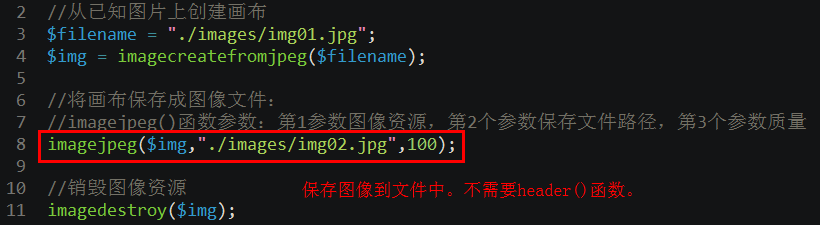




## 2、输出图像到浏览器或保存文件imagejpeg()

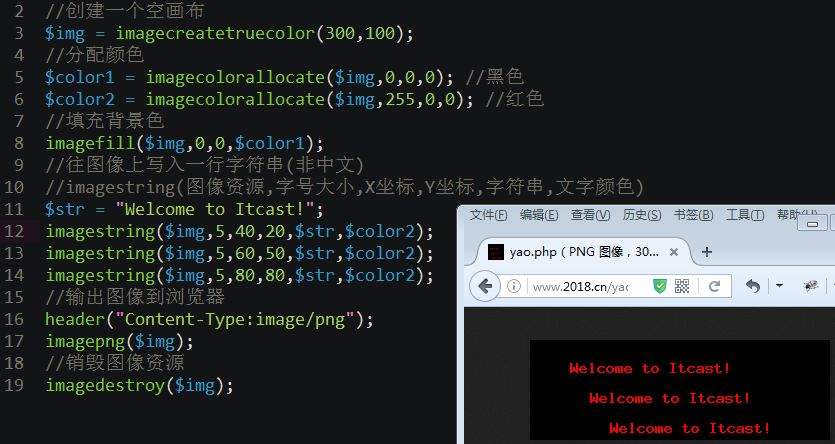
* 描述：以 JPG/GIF/PNG 格式将图像输出到浏览器或文件
* 语法：**bool imagejpeg ( resource $image [, string $filename [, int $quality ]] )**
* 参数：
  + $filename，将创建的图像保存到文件；如果省略，则直接在浏览器输出。
  + 如果要省略这个参数而提供 quality 参数，使用NULL。
  + quality 为可选项，范围从 0（最差质量，文件更小）到 100（最佳质量，文件最大）。默认的质量值（大约 75）。
* 提示：imagegif()、imagepng()，与imagejpeg()格式一样，但没有第3个参数。





## 3、水平地画一行字符串imagestring()

* 描述：水平地画一行字符串
* 语法：**bool imagestring(resource $img,int $font,int $x,int $y,string $s,int $col)**
* 参数：
  + $img 图像资源；
  + $font字体大小，取值1、2、3、4、5，使用内置字体；
  + $x , $y 绘制字符串的开始坐标，一般在字符串左上角；
  + $s 代表要绘制的一行字符串；
  + $col 代表文本颜色。
  + $s，代表一行字符串；$col，代表文本颜色；



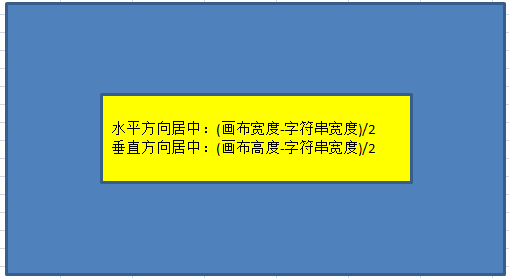
## 4、获取画布的宽度和高度

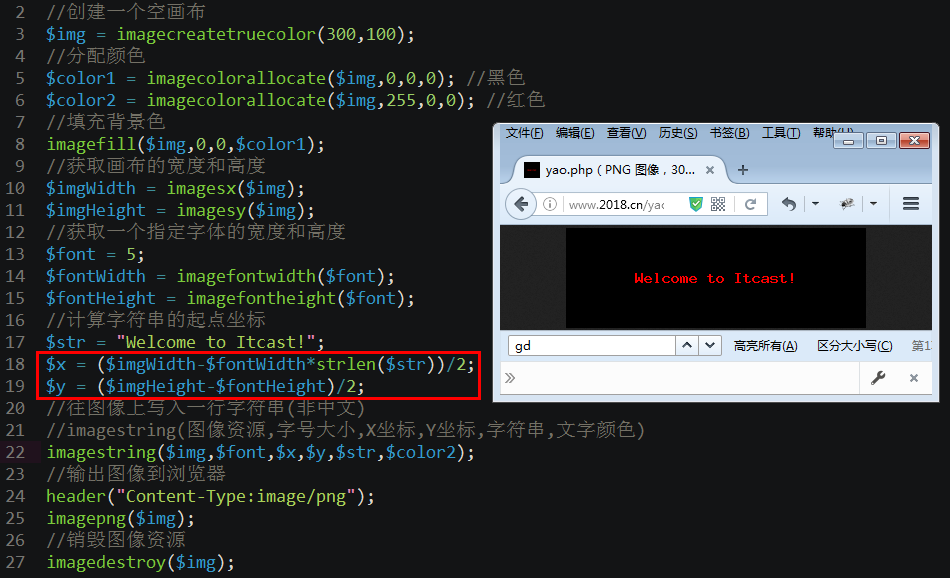
* 宽度：int imagesx ( resource $image )
* 高度：int imagesy ( resource $image )

## 5、获取内置字体的宽度和高度

* 描述：返回指定字体一个字符宽度或高度的像素值。
* 字体宽度：**int imagefontwidth ( int $font )**
* 字体高度：**int imagefontheight ( int $font )**
* 提示：$font为字体大小，取值1-5，最大为5。

## 6、实例：在图像上绘制一行居中的字符串





## 7、画一矩形并填充

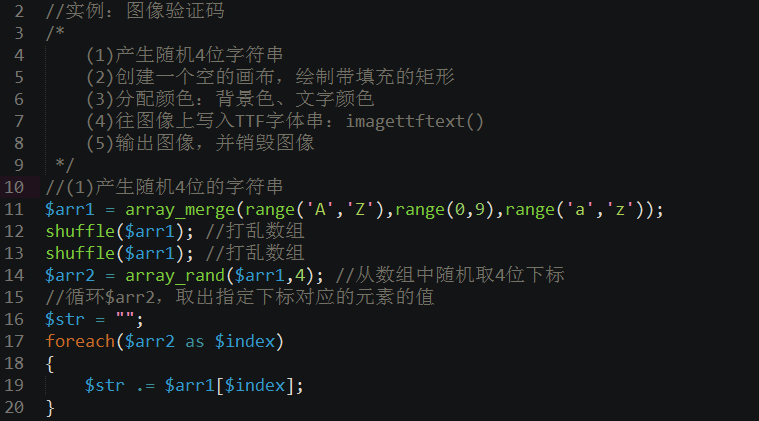
* 描述：画一矩形并填充
* 语法：**bool imagefilledrectangle ( resource $image , int $x1 , int $y1 , int $x2 , int $y2 , int $color )**
* 参数：
  + $x1 , $y1 左上角坐标；
  + $x2 , $y2 右上角坐标；
  + $color 填充背景色。

## 8、画一个单一像素

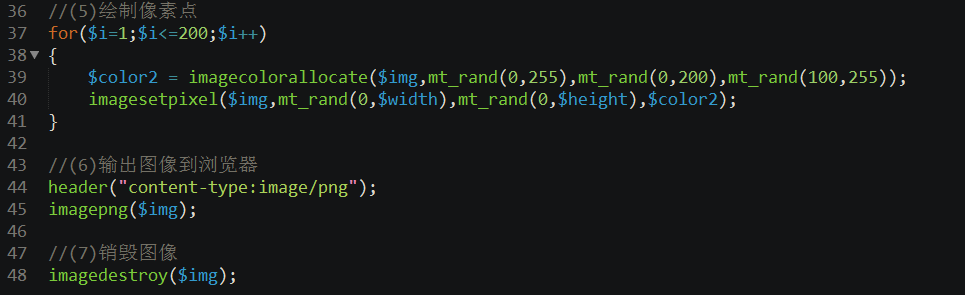
* 描述：画一个单一像素
* 语法：**bool imagesetpixel ( resource $image , int $x , int $y , int $color )**
* 说明：imagesetpixel() 在 image图像中用 color颜色在 x，y 坐标（图像左上角为 0，0）上画一个点。

# 实例：图像验证码

## 1、绘制图像验证码









## 2、产生一个指定范围的数组range()

* 描述：建立一个包含指定范围单元的数组
* 语法：**array range ( mixed $start , mixed $limit [, number $step = 1 ] )**
* 参数：
  + $start 指定范围第1个值；
  + $limit 指定范围最后1个值；
  + $step 指定步长值，必须为正整数，默认为1。

## 3、合并数组array\_merge()

* 描述：合并一个或多个数组
* 语法：**array array\_merge ( array $array1 [, array $... ] )**
* 说明：如果输入的数组中有相同的字符串键名，则该键名后面的值将覆盖前一个值。然而，如果数组包含数字键名，后面的值将不会覆盖原来的值，而是附加到后面。

## 4、从数组中随机取出一个或多个单元(下标)

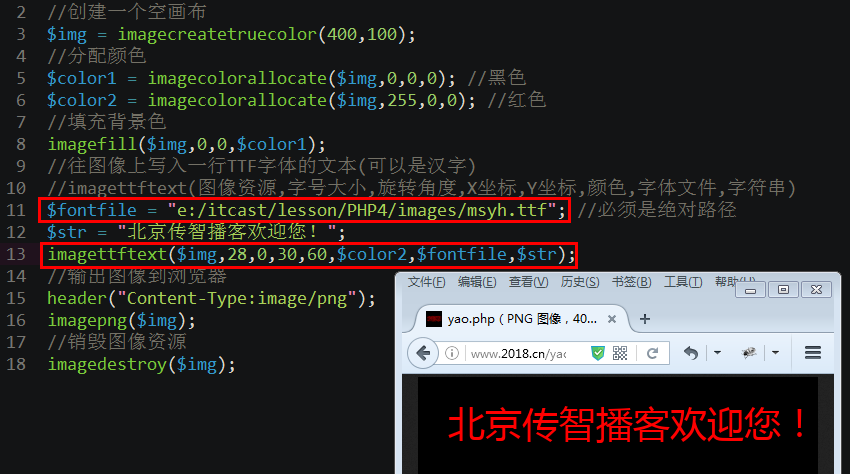
* 描述：从数组中随机取出一个或多个单元
* 语法：**mixed array\_rand ( array $input [, int $num\_req = 1 ] )**
* 参数：$input代表当前数组 , $num\_req指明了你想取出多少个单元。

## 5、生成更好的随机数

* 描述：生成更好的随机数
* 语法：**int mt\_rand ( int $min , int $max )**
* 参数：min可选的、返回的最小值（默认：0）；max可选的、返回的最大值。

# 实例：往图像上写入一行汉字

* 描述：用 TrueType 字体向图像写入文本
* 语法：**array imagettftext ( resource $image , float $size , float $angle , int $x , int $y , int $color , string $fontfile , string $text )**
* 参数：
  + $size，字号大小，自定义，同word字号一样；
  + $angle，旋转角度(0~360)；
  + $x和$y，定义了第一个字符的基本点（大概是字符的左下角）。
  + $fontfile，是想要使用的 TrueType 字体的**绝对路径**；
  + $text，UTF-8 编码的文本字符串(其它编码要进行转换)。



# 实例：制作图像水印效果

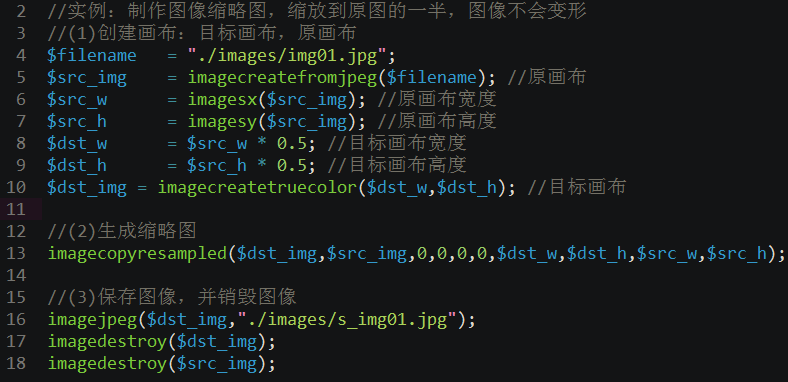
* 描述：为图像分配透明颜色imagecolorallocatealpha()
* 语法：**int imagecolorallocatealpha ( resource $image , int $red , int $green , int $blue , int $alpha )**
* 说明：imagecolorallocatealpha() 的行为和 imagecolorallocate() 相同，但多了一个额外的透明度参数 alpha，其值从 0 到 127。0表示完全不透明，127 表示完全透明。





# 实例：生成图像缩略图

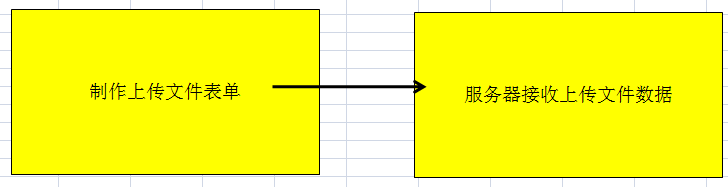
* 描述：将一幅图像中的一块正方形区域拷贝到另一个图像中，平滑地插入像素值，因此，减小了图像的大小而仍然保持了极大的清晰度。
* 语法：**bool imagecopyresampled ( resource $dst\_image , resource $src\_image , int $dst\_x , int $dst\_y , int $src\_x , int $src\_y , int $dst\_w , int $dst\_h , int $src\_w , int $src\_h )**
* 参数：
  + $dst\_image，目标图像；
  + $src\_image，源像图；
  + $dst\_x和$dst\_y，目标图像x、y坐标；
  + $src\_x和$src\_y，源图像x、y坐标；
  + $dst\_w和$dst\_h，目标图像的宽度和高度；
  + $src\_w和$src\_h，源图像的宽度和高度；
* 提示：如果源和目标的宽度和高度不同，则会进行相应的图像收缩和拉伸。

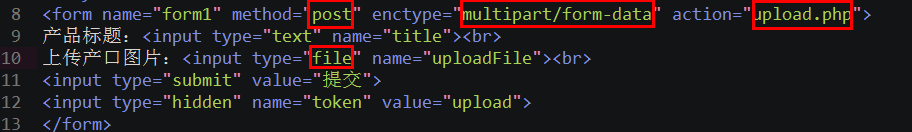


# 文件上传

## 1、文件上传原理

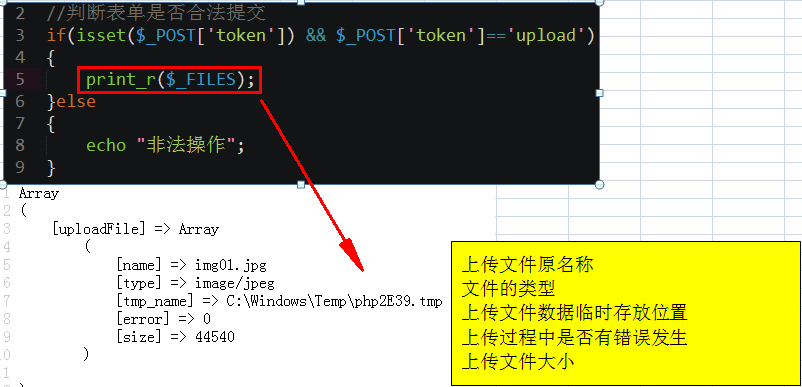
* 上传文件，就是将文件从浏览器端传到服务器端；
* 上传文件，必须使用<form>标记来向服务器端发数据；
* 上传文件，<form>标记的method属性值必须是POST；
* 上传文件，<form>标记的enctype属性值必须是multipart/form-data；
* 上传文件，必须使用<input type = ‘file’ name=’upload’>标记实现。



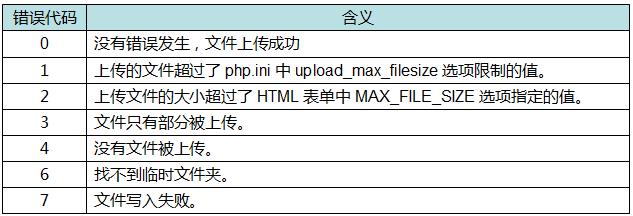


## 2、超全局变量数组 $\_FILES

* $\_POST数组中保存的是普通表单元素数据；
* $\_FILES数组中保存的是上传文件的信息。



## 3、上传文件错误代码



提示：0、1、4比较重要。

## 4、查看上传的临时文件位置

* 上传文件的默认临时目录，如果没有指定php.ini配置项upload\_tmp\_dir的值，则使用操作系统临时目录(c:\windows\temp)。通过phpinfo()函数查看。
* 为什么在 c:\windows\temp 中看不到上专文件呢？因为临时文件是短暂存在的，也就是在脚本执行完毕后，就消失了。



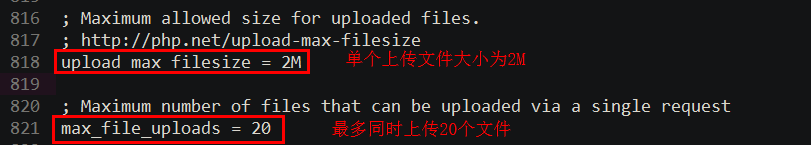
## 5、将上传文件移动到新位置

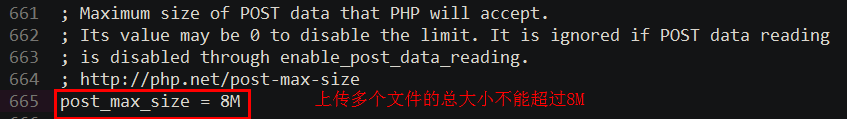
* 描述：本函数检查并确保由 filename 指定的文件是合法的上传文件（即通过 PHP 的 HTTP POST 上传机制所上传的）。如果文件合法，则将其移动为由 destination 指定的文件。最好在临时文件没有消失前移动。
* 语法：**bool move\_uploaded\_file ( string $filename , string $destination )**
* 参数：$filename，指定上传的临时文件名；$destination，指定新的文件名路径；
* 注意：如果文件已经存在，则会覆盖操作。



## 6、上传文件的相关配置(php.ini)

* upload\_max\_filesize配置：上传单个文件的大小限制，默认为2MB。
* post\_max\_size配置：规定上传多个文件的总大小，默认为8MB。
* max\_file\_uploads配置：规定最多上传的文件个数，默认为20个。





## 7、获取文件路径信息 pathinfo()

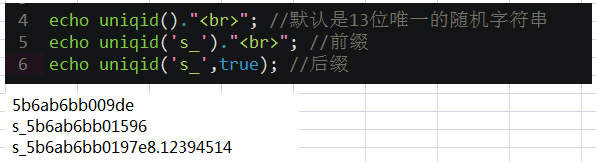
* 描述：返回文件路径的信息 ;
* 语法：**mixed pathinfo ( string $path [, int $options = PATHINFO\_DIRNAME | PATHINFO\_BASENAME | PATHINFO\_EXTENSION | PATHINFO\_FILENAME ] )**
* 参数：
  + $path，要解析的路径。
  + $options，如果省略，返回全部单元；
  + PATHINFO\_DIRNAME，目录名称；
  + PATHINFO\_BASENAME，文件名称；
  + PATHINFO\_EXTENSION，扩展名；
  + PATH\_FILENAME，文件名。

## 8、检查数组中是否存在某个值 in\_array()

* 描述：检查数组中是否存在某个值
* 语法：**bool in\_array ( mixed $needle , array $arr )**
* 参数：$needle检索的值，$arr原数组。

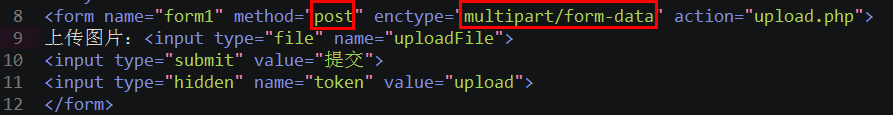
## 9、生成唯一ID uniqid()

* 描述：生成一个唯一ID
* 语法：**string uniqid ([ string $prefix = "" [, bool $more\_entropy = false ]] )**
* 参数：
  + $prefix前缀字符串，如果省略，返回字符串长度为13；
  + $more\_entropy后缀字符串，如果为true，返回字符串长度为23。



# 实例：单个文件上传

## 1、上传表单制作upload.html



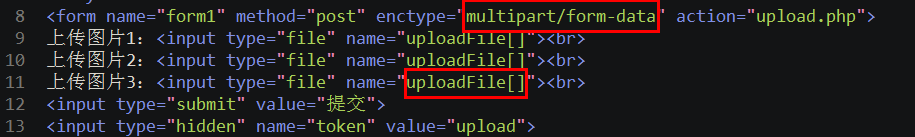
## 2、上传文件的程序处理upload.php





# 实例：多个文件上传

## 1、制作上传表单



## 2、上传多个文件的程序处理

