目 录

[PHP中的时间函数 3](#_Toc22189)

[时间戳的了解 3](#_Toc6219)

[设置时区的方法 3](#_Toc16921)

[与时间相关的函数 4](#_Toc2)

[php中的错误异常处理 5](#_Toc4724)

[认识错误 5](#_Toc19177)

[错误的级别 5](#_Toc16220)

[防止错误 5](#_Toc21309)

[自定义处理错误的方式 6](#_Toc9188)

[PHP中的文件处理 6](#_Toc11584)

[文件系统了解 6](#_Toc32424)

[文件相关的函数 6](#_Toc25512)

[文件的打开关闭以及读取 7](#_Toc1003)

[文件的写入 8](#_Toc25742)

[PHP中的目录操作 9](#_Toc12503)

[目录操作的相关函数 9](#_Toc10773)

[PHP中文件上传与文件下载 10](#_Toc27136)

[文件上传 10](#_Toc5501)

[上传原理 10](#_Toc18106)

[上传需注意 10](#_Toc30615)

[上传中需考虑 11](#_Toc14389)

[上传到临时系统文件中错误号 11](#_Toc23019)

[文件下载 11](#_Toc29321)

[PHP中的正则表达式 12](#_Toc30235)

[简单介绍正则表达式及其作用 12](#_Toc28439)

[正则表达式的语法 12](#_Toc30015)

[正则定界符 13](#_Toc23266)

[正则表达式的原子 13](#_Toc8021)

[正则表达式中的元字符 13](#_Toc6357)

[正则表达式模拟修正符 14](#_Toc8953)

[正则表达式的函数 15](#_Toc16967)

[PHP中的图像处理 16](#_Toc2017)

[GD库图像处理 16](#_Toc24065)

[GD2程序绘图 16](#_Toc17926)

[GD2相关函数 16](#_Toc28982)

[创建 16](#_Toc7836)

[分配颜色 17](#_Toc16250)

[绘图操作 17](#_Toc32731)

[通知 17](#_Toc15161)

[显示/保存 17](#_Toc9040)

[释放 18](#_Toc11628)

[总结 18](#_Toc21795)

# 

# PHP中的时间函数

## 时间戳的了解

* 计算机统一时间的方式是使用unix时间戳
* 从1970年1月1日0时0分0秒开始到指定时间的秒数，时间戳属于整型数据类型（该时间为计算机的元年）
* php的时间戳是从1970年开始最大表示到2038年1月27日左右

## 设置时区的方法

服务器默认的时间都是国家统一时间，格林威治。所以在使用服务器前需要设置当前服务器的时区。位于用于指定的时区。

* 修改配置文件->英文法

找到：php.ini配置文件

查找：date.timezone选型

找到：php手册中-所支持的时区列表

php手册-》目录-》函数参数-》时间与日期扩展-》所支持的时区列表-》亚洲

设置选项：date.timezone  = Asia/Shanghai（大洲/城市）

重启：apache服务器方可生效

注意：必须在支持的时区列表中选取。

* 修改配置文件->编码法

修改：php.ini 配置文件

找到：date.timezone= ETC/GMT-8

重启：apache服务器

* 修改配置文件->缩写法

修改：php.ini 配置文件中

找到：date.timezone = PRC（中华人民共和国）

重启：apache服务器

* 使用函数修改法

使用函数修改当前页面的时区

使用date\_default\_timezone\_set() 函数来设置时区，其参数和配置文 件 中的date.timezone 的值一样。

现实中使用该方法，会在每个页面 的开始部门包含并且运行。

## 与时间相关的函数

* time() 可以获取当前时间戳

@param1 无

返回值：返回当前时间戳

* date() 格式化一个时间戳（转换成看的懂的时间格式）

@param1 要转换时间格式的字符串

* 常用格式：

Y 四位数字的年份

m 两位数字的月份

d  两位数字的天数

H  24小时 00-23

i   两位数的分钟

s   两位的秒数

w  星期中的第几天

t   给定月份有多少天

注意：如还需要其他格式自行查手册

@param2 可选参数，如果指定一个时间戳，表示格式化这个时间戳，如 果没有指定时间戳，默认是当前时间

返回值：返回按照给定格式的字符串

* mktime()  获得一个日期的时间戳（给定一个日期，返回一个时间戳）

@param1 可选参数

参数从左到右：时、分、秒、月、日、年

参数可以从右向左省略，任何省略的参数都会被计算是当前时间值

返回值：返回指定日期的时间戳

* microtime()  返回当前unix时间戳和微妙数

@param1：可选参数

如果设置为true，表示返回时间戳.微妙数

默认是false，表示返回微妙数（空格）时间戳

* sleep()  睡觉

@param1  睡多长时间

* strtotime()  将任何英文文本格式时间转换成时间戳

@parma1 日期/时间字符串（用now字符串来获取当前时间戳，其他带数字正负的，都表示日期的前几天或者后几天）

@param2 可选参数 用来计算返回值的时间戳

返回值：成功则返回时间戳 否则返回false

注意：任何函数都不可能一个一个的罗列出来，笔记中整理的都是与之相关常用的函数，最重要的还是要查手册。

# php中的错误异常处理

## 认识错误

* E\_ALL -> 所有的错误
* E\_ERROR -> 致命的运行错误
* E\_WARNING -> 警告级别的错误
* E\_NOTICE -> 提示级别的错误
* E\_STRICT -> 编码标准化警告
* E\_CORE\_ERROR ->  php启动时初始化过程中的错误（php没有启动）
* E\_CORE\_WARNING ->  php启动时初始化过程中的警告
* E\_COMPILE\_ERROR -> 编译时的致命错误
* E\_COMPILE\_WARNING -> 编译时的警告错误
* E\_USER\_ERROR -> 用户自定义的致命错误
* E\_USER\_WARNING -> 用户自定义的警告错误
* E\_USER\_NOTICE  -> 用户自定义的提示

## 错误的级别

* Error：  致命错误，会终止程序的运行
* Warning： 警告级别的错误，不会终止程序的运行
* Notice：  提示 提醒（算不上错误）

## 防止错误

* 让不该看到的人看不到错误
* 使用错误抑制符@，只能屏蔽单行错误而且效率低。
* 修改php.ini 配置文件中的display\_errors选项，设置on显示错误，设置off不显示错误
* 修改php配置文件中error\_reporting选项，可以定义任意类型的错误输出或者不输出

  E\_ALL  &（并且） ~（除了某一个）

E\_ALL &~E\_NOTICE 报所有的错误但是除了E\_NOTICE

* 使用error\_reporting() 函数，可以和配置文件中的error\_reporting选项实现相同效果，设置的值也是一样
* 让该看到的人看到错误（制作网站的相关人员）
* 第一步:关闭配置文件中display\_errors 设置为off
* 第二步:设置存储错误的类型，设置配置文件中error\_reporting选项，指定需要存储的错误类型
* 第三步:开启错误日志记录开关，设置配置文件中的log\_errors =On
* 第四步:设置值错误日志记录的地点
* error\_log = syslog 记录错误到系统日志中(打开去掉分号)
* 系统日志查找方式-》开始-》控制面板-》管理工具-》事件查看器-》windows日志-》应用程序
* 从控制面板中查看错误看着比较难分辨，这是我们可以自定义写入指定的文件中（error\_log = syslog前面的分号在加上）
* error\_log = “写入错误文件的路径”
* 注意：error\_log不会帮我们自定义创建目录，只能帮我们自动创建文件。

注意：要知道一切通过配置文件修改的东西，都可以通过函数来修改。

## 自定义处理错误的方式

* set\_error\_handler(); 设置一个用户定义的错误处理函数

@param1：传入一个回调函数

@param2：可选参数，设置错误的类型

注意：用户自定义的回调函数需要接受四个参数。

错误号

错误内容

错误文件

当前错误行

* file\_put\_contents() 将内容写入指定的文件中

@param1：要写入文件的地址

@param2：要写入的内容

@param3：可选参数。设置为FILE\_APPEND追加数据而不覆盖

返回值：返回写入的字节数

注意：该函数默认以替换的方式写入文件。

如果文件存在，则清空文件后写入。如果文件不存在，创建文件后写入 文件。

# PHP中的文件处理

## 文件系统了解

对所有的文件进行任何操作都需要文件系统相关函数

学习文件系统就是学习操作文件相关的函数

## 文件相关的函数

* touch()  创建文件

@param1 要创建或者说要修改访问和修改时间的文件名

@param2 可选参数，修改指定的文件的修改时间，如果没有设置则使用 系统时间

@param3  可选参数，修改指定文件的访问时间，如果没有设置则使用 系统时间。

返回值：成功返回true，失败返回false

注意：

* touch() 本身的功能是：设定文件的访问和修改时间
* 如果该文件不存在，则会自动创建文件。
* 我们借用touch() 函数帮我们创建文件

* rename()  移动文件/还有重命名文件的功能

@param1  要移动文件的原路径

@param2 要移动文件的目标路径

返回值：成功返回true，失败返回false

* copy()  复制文件

@param1  要复制文件的原路径

@param2 要复制文件的目标路径

返回值：成功返回true  失败返回false

* unlink()  删除文件

@param1 要删除的文件名

返回值：成功返回true，失败返回false

* filesize()  获取文件大小

@param1 要获取大小的文件名

返回值：返回文件的大小字节数，失败返回false

* is\_file()  判断是否是文件

@param1 要判断是否是文件的文件名

返回值：是文件则返回true，不是文件则返回false

* file\_exists()  判断文件或者目录是否存在

@param1 要判断是否存在的文件或者目录名称

返回值：如果存在则返回true，不存在返回false

## 文件的打开关闭以及读取

* fopen() 打开文件或者url

@param1 要打开文件的路径

@param2 打开文件的方式

* r 只读方式打开文件
* r+ 读写方式打开文件(可以执行部分写入功能)
* w 写入方式打开文件

注意：如果文件不存在则创建文件后在写入文件。

如果文件存在清空文件内容后在写入文件。

* w+ 读写方式打开文件（可以执行部分的读取功能）
* a  累加或者追加方式打开文件

注意：如果文件不存在则创建文件后在写入文件。

如果文件存在则追加数据而不是覆盖。

* a+增加或者追加的模式打开文件（可执行部分读取功能）
* x 异或方式打开

异或的特点：相同为假，不同为真。

文件存在就报错，不存在新建并写入

* x+ 增加异或模式（可以执行部分读取功能）

返回值：成功打开文件资源，失败返回false

* fclose() 关闭文件

@param1  有fopen() 成功打开文件返回的资源

返回值：成功返回true，失败返回false

* fread()  读取文件

@param1 由fopen() 成功打开文件返回的资源

@param2 每次读取的字节数

返回值：返回读取到的内容，失败返回false

注意：在gb字符集系统中一个中文站位2个字节

  在utf-8字符集中一个中文汉字站位3个字节

* feof()  测试文件指针是否到了文件结尾处

@param1 由fopen() 成功打开文件返回的资源

返回值：如果文件指针到了结尾或者出错则返回true 没到结尾返回false

* file\_get\_contents() 快捷读取文件

@param1  要读取文件的路径

返回值：返回读取到的内容

## 文件的写入

* fwrite()  写入文件函数

@param1 由fopen() 成功打开文件返回的资源

@param2 要写入文件的内容

@param3 可选参数，设置写入文件的字节数

返回值：返回写入文件的字节数

* file\_put\_contents() 快捷方式写入文件

@param1  要写入的文件地址，如果文件不存在则自动创建文件后在写入

@param2 要写入文件的内容

@param3 可选参数，如果设置为FILE\_APPEND表示追加数据的方式写入 文件

注意：该函数默认是w方式写入文件

返回值：返回写入文件的字节数。

# PHP中的目录操作

## 目录操作的相关函数

* opendir();   打开目录

@param1  要打开目录的路劲

返回值：成功返回资源，失败返回false

* readdir();   读取目录

@param1 由opendir打卡目录成功返回的资源

返回值：读取成功返回读取的目录名称，读取失败或者读取到目录最后 时返回false

注意：每个目录下默认都有一个点和一个点点目录

读取目录时，第一次读取会读到.（点）。点代表当前目录

读取目录时，第二次读取会读取到..（点点）。点点代表上一级目录

* closedir(); 关闭目录

@param1 由opendir打卡目录成功返回的资源

* mkdir()  创建目录

@param1  要创建的目录名称

@param2 可选参数，表示设置创建目录的权限   0777表示最高权限

@param3  可选参数，如果设置true 表示允许镶嵌的方式创建目录，默认 为false

返回值：成功返回true，失败返回false

* rmdir() 删除目录

@param1 要删除目录的路径

返回值：成功返回true 失败返回false

注意：只能删除一个空目录，如果想正常删除目录操作，需要自己写功能

* dirname();  返回路径中目录部分

@param1  一个路径

返回值：返回路径中目录部分

* basename();  返回路径中文件部分

@param1 一个路径

返回值: 返回路径中文件部分

* pathinfo()  获取路径信息

@param1  一个路径

返回值：返回一个数组，数组中包含，目录部分，文件部分，文件后缀，文件名字。

# PHP中文件上传与文件下载

## 文件上传

### 上传原理

将本地的文件复制一份传送到网络的服务器中储存

* 通过浏览器选择本地文件
* 通过互联网传送到网络服务器中的临时目录

（系统的临时目录temp格式，其实就是垃圾目录，为了防止病毒，系统会自动将上传的文件随机改名后缀为.tmp。这个文件是不可以运行的文件，当提交的php脚本运行结束后，如果没有移动走该临时文件，那么系统的临时文件就会自动删除该文件）

* 如果上传到临时目录中没有出错，则把文件从临时目录中移动到www目录下将其改好名字。即为实现文件的上传

### 上传需注意

* HTML表单部分
* 上传文件时，传输方式必须为post方式
* 需要指定表单输出编码格式：enctype="multipart/form-data"
* PHP页面部分

如果是文件上传，那么上传的文件在php脚本中不能用$\_POST来接受， 必须要用$\_FILES来接受，其他内容依然用$\_POST来接受

* PHP配置文件相关的选项
* upload\_max\_filesize = 2M  允许上传文件大小的最大值设置
* file\_uploads = On  允许文件上传的设置
* upload\_tmp\_dir ="c:/wamp64/tmp" 文件上传的临时目录的设置
* max\_file\_uploads = 20 允许最大上传文件个数的设置
* post\_max\_size = 8M  post最大传输内容的最大值设置

注意：upload\_max\_filesize 设置的值不允许超过post\_max\_size设置的值

### 上传中需考虑

* 判断是否上传成功，判断错误号，是否从客户端上传到系统临时目录过程中成功
* 判断上传文件的类型是否符合要求
* 判断上传文件的大小时候符合要求
* 创建上传成功后保存的目录
* 上传上来的图片不能重名  重新命名
* 移动文件（从系统临时文件目录中移动到指定的目录中并且图片要求重命名）

### 上传到临时系统文件中错误号

可查看手册post上传方法来整理

文件的类型：可查看MIME类型

文件移动时：

move\_uploaded\_file() 将长传的文件移动到新位置

@param1 原路径

@param2 需要移动的路径

返回值：成功返回true 失败返回false

die(); 与break;

函数中不要用以上两位函数，用return代替

## 文件下载

就是将服务器中的文件复制一份保存到本地即为下载

* 下载需要的三行代码

//声明要下载文件的类型

header('Content-type: image/jpeg');

//对下载文件进行描述并且制定下载文件的名称

header('Content-Disposition:attachment;filename="下载文件的名称.jpg");

//读取需要下载的文件

readfile('./文件目录/文件名.后缀');

* IE浏览器中编码问题

iconv();  字符集按要求的字符编码来转换

@param1 原字符集

@param2 需要转换成那种字符集

@param3 那个字符串需要转换

返回值：返回转换后的字符串 失败返回false

* 设置页面字符集

header('Content-type:text/html;charset=utf-8');

使用header做条件注意：

尽量不要在header前面输出，有输出不会报错，但是尽量不要这么做，因为不符合常理，在php版本小于5.3之前不允许header前面有输出

注意：header的功能有很多需要一一了解

header跳转

header('location:需要跳转的路径');

# PHP中的正则表达式

## 简单介绍正则表达式及其作用

* 正则表达式的了解
* 正则表达式就是描述字符串排列模式的一种自定义语言规则
* 如果可以使用字符串处理函数完成的任务，就不要使用正则表达式
* 有一些复杂性的操作，只能用正则表达式
* 正则表达式也称为一种模式表达式
* 正则表达式就是通过具有特定规则的模式，在与输入的字符串信息进行比较、分割、查找、替换等操作。
* 正则表达式的特点
* 正则表达式也是一个字符串
* 由具有特殊意义的字符串组成的字符串
* 具有一定编写规则，也是一种模式
* 看做是一种编程语言：是用一些特殊字符，按照规则编写出来一个字符串，形成一种模式

注意：如果正则表达式不和函数一起使用，则就是一个字符串，如果正则表达式放到某个函数中使用，才能发挥正则表达式的真正作用。

* PHP中提供了两套正则表达式函数库
* Posix扩展正则表达式 ereg\_开头的函数
* perl兼容正则表达式 preg\_开头的函数
* 推荐使用prel 兼容正则表达式函数库

## 正则表达式的语法

* 正则定界符
* 正则表达式中的原子
* 正则表达式中的元字符
* 正则表达式的模式修正符

### 正则定界符

正则的定界符就是用来声明正则表达式边界的符号。

正则表达式常用的边界符为 /，实际上任何非数字字母以及反斜线 \ 之外的任意字符都可以做正则表达式的边界符。

注意：一个完整的正则表达式有两个边界符

* 一个是字符串类型的引号边界符  ''
* 一个是正则表达式的边界符 //(一般用/)
* 整体就是 '//'

### 正则表达式的原子

 组成正则表达式的最小单位就是原子

正则的数字字母都是原子，大部分不可见的字符也是原子。

例如：\n\r\t...这些转译字符都是元素

* 正则中原子
* \d 表示 0-9 之间的一个数字
* \D 表示除了0-9之间的任意一个字符
* \s 表示匹配任意一个空白字符包括 \n\r\t..
* \S  表示除了空白符或者转译字符之外的任意一个字符
* \w 表示大小写字符数字以及下划线当中任意一个字符
* \W 表示除了大小写字母数字以及下划线之外的任意一个字符

### 正则表达式中的元字符（原子修饰符）

* [] 原子列表

    表示在指定的字符中选择一个原子

  例如:[5-9] 表示5 6 7 8 9连续得字符允许缩写为：[开始字符 - 结束字符]

* [^] 排除列表

例如：[^4 ] 表示除了4之外的任意一个字符都可以

* 用元字符（原子修饰符）来模拟原子

\d  [0-9]   \D [^0-9]

\s [\n\r\t]  \S[^\n\r\t]

\w[a-zA-Z0-9]  \W[^a-zA-Z0-9]

* 原子数量修饰符
* ?  表示前面的原子出现0次或者1次
* + 表示前面的原子出现1次或者多次
* \*  表示前面的原子出现0次或者多次
* 原子数量修饰列表
* {m,n} 表示前面的原子最小出现m次，最多出现n次
* {m,}表示前面的原子最少出现m次
* {m} 表示前面的原子只能出现m次
* 原子数量修饰列表模拟数量修饰符

？ {0,1}

+ {1,}

\* {0,}

* 字符串边界修饰符
* ^ 表示以指定字符开头的内容
* $ 表示以指定字符结尾的内容

注意：

* ^ $一前一后表示精准修饰限定两者之间的内容是否一致
* 如果要使用^和$ 可以匹配每一行的开始和结束部分，需要使用模拟修正符m

* | 选择修饰符
* 或与且类似
* () 模式单元
* 改变优先级
* 将多原子视为一个原子，可以使用原子数量修饰符
* 将匹配到的括号中的内容暂时存为内存中
* 如果使用() 仅仅为了模拟一个原子使用原子数量修饰符，那么可以在括号的开始处使用?: 来接触存于内容的作用，提高效率
* 可以将括号的内容在函数的辅助下进行反向引用操作

* 元字符
* .（点）匹配除了\n之外的任意一个字符

### 正则表达式模拟修正符

* i 忽略大小写
* x 忽略正则表达式中得空白
* s 使用.(元字符)可以匹配到\n
* S 加速匹配
* U 模式  改变贪婪模式

改变贪婪模式，程序默认是贪婪模式，匹配最远的结束为止，我们可以在正则中使用\*?匹配任意字符的同时使用非贪婪模式

注意：如果此时程序为贪婪模式，使用U后，则程序变为非贪婪模式。 如果此时程序为非贪婪模式，使用U后，程序变为贪婪模式

## 正则表达式的函数

* preg\_grep()  返回与模式匹配的数组单元

@param1：正则表达式

@param2：要匹配的数组

@param3：可选参数 默认为false。设置为true表示返回与模式匹配相 反的数组单元

返回值：返回与模式匹配的数组单元

* preg\_match() 进行一次正则匹配 适合做查询有没有

@param1： 正则表达式

@param2： 要匹配的字符串

@param3： 可选参数，返回与正则表达式匹配的内容

返回值：成功返回1，失败返回0   1表示匹配的次数

* preg\_match\_all() 进行正则匹配  适合查询有多少

@param1： 正则表达式

@param2： 要匹配的字符串

@param3： 可选参数，返回与正则表达式匹配的内容

返回值：返回匹配的次数，如果没有匹配带则返回0

* preg\_split()  用正则表达式分隔字符串

@param1：正则表达式

@param2： 要分隔的字符串

@param3： 返回分隔后的数组

* preg\_replace() 执行正则表达式的替换

@param1：要查找正则表达式

@param2：要替换的内容

@param3：在哪里进行替换操作（数组还是字符串）

返回值：返回替换后的数组或者字符串

新闻采集器的介绍与原理

后期补充

# PHP中的图像处理

## GD库图像处理

GD库是[php](https://baike.baidu.com/item/php" \t "https://baike.baidu.com/item/GD%E5%BA%93/_blank)处理图形的扩展库，GD库提供了一系列用来处理图片的[API](https://baike.baidu.com/item/API" \t "https://baike.baidu.com/item/GD%E5%BA%93/_blank)，使用GD库可以处理图片，或者生成图片，也可以给图片加水印。

* 那些地方需要使用图像的处理
* 验证码
* 相册中处理
* 在线图片生成器

* 配置GD2库

确定当前环境要支持GD2相关操作

* 查看phpinfo中是否存在gd模块的说明
* 如果没有查询到gd模块，需要修改配置文件中的extension=php\_gd2.dll 去掉前面的分号，然后重启apache
* 如果修改了配置文件还不能开启GD2模块，需要查看环境中PHP文件中ext文件夹是否具有php\_gd2.dll文件的存在，如果没有需要在官网下载。

## GD2程序绘图

* 步骤
* 创建画布
* 为每一副画分配颜色
* 使用相关函数进行绘图操作（圈、点、线）
* 通知浏览器图片的相关信息（告诉浏览器创建图片的类型 jpeg jpg png gif）
* 将图片显示到浏览器或者保存到文件中
* 释放资源

## GD2相关函数

### 创建

* imagecreate() 创建一个基于调色板的画布（颜色少，只能表示出256种颜色 适合做GIF图片）

@param1：图片的宽度

@param2：图片的高度

返回值：成功返回资源，失败返回false

* imagecreatetruecolor() 创建真彩色图片（颜色65535种颜色，适合做jpg/png）

@param1：图片的宽度

@param2：图片的高度

返回值：成功返回资源，失败返回false

注意：如果使用真彩色创建图片，必须使用imagefill填充背景

### 分配颜色

* imagecolorallocate()分配颜色

@param1：要分配颜色的图片资源

@param2：红色 0-255

@param3：绿色 0-255

@param4：蓝色 0-255

返回值：返回颜色标示

### 绘图操作

* imagefill()区域填充

@param1：要填充的资源

@param2：填充的X坐标

@param3：填充的Y坐标

@param4：要填充的颜色

通过函数来完成画圆、画弧线、画点、画线等等一系列绘图操作（具体函数可查考手册）

### 通知

* header()告诉浏览器图片相关信息

header('Content-type:image/jpeg');

注意：图片类型为自己定义常用有：image/jpeg  image/png image/gif

### 显示/保存

* imagejpeg() 输出jpeg、jpg图片到浏览器或者保存到本地

@param1： 要输出的图片资源

@param2： 可选参数 如何设置表示图像保存的路径

返回值： 布尔值

* imagepng() 输出png图片到浏览器或者保存到本地

@param1： 要输出的图片资源

@param2： 可选参数 如何设置表示图像保存的路径

返回值： 布尔值

* imagegif() 输出gif图像到浏览器或者保存到本地

@param1： 要输出的图片资源

@param2： 可选参数 如何设置表示图像保存的路径

返回值： 布尔值

注意：告诉浏览器图片是什么类型后下面就应对图片的类型选择对应的函数输出到浏览器或者保存图片

### 释放

* imagedestroy() 释放资源 销毁图像

@param1：图片资源

返回值：返回布尔值

注意：有几张图片最后都要依依释放图片的资源。

# 总结

在PHP功能模块这一章中不需要向之前PHP基础篇中需要深刻的记忆中弄懂功能模块的特定的函数，重点在于各个模块的思路与方法。因为PHP中系统的函数有很多是记不完的，需要用时查手册就可以了。

说道手册的查看也是很有讲究的，为一个函数的后面手册中都有好几个例子，供我们学习。