

## 第19章 PHP与图像处理

- PHP除了用来创建Web应用之外，还可以生成图片或对图片进行加工处理。通过一个名叫GD的PHP扩展库，PHP可以动态生成不同格式图像、绘制图线、对图片加以处理等。由于有GD的强大支持，PHP的图像处理功能可以算是PHP的一个强项，它简捷易用，功能强大。甚至可以完成像素处理、颜色转换、灰度变换等高级功能。本章就从PHP的图像处理函数开始，向读者介绍如何通过PHP完成图像处理。



## 19.1 PHP的图像处理库——GD库

- 可以创建及操作多种不同图像格式的图像文件，包括gif, png, jpg, bmp等。更方便的是，PHP可以直接将图像流输出到浏览器。使用PHP处理图像通常情况下需要GD扩展库的支持，对于Linux/UNIX用户来说，需要在编译PHP时加上图像函数的GD库。GD和PHP 还可能需要其它的库，这取决于需要支持哪些图像格式而定。从PHP4.3，PHP起绑定了一个GD库的版本，目前是GD2。本书就使用GD2来介绍PHP图像处理函数。



## 19.1.1 安装GD库

- 对Linux/UNIX用户来说，要激活GD支持，配置PHP时需要加上`--with-gd[=DIR]`，其中DIR是GD的安装目录。如果要使用推荐绑定的GD库版本，可以使用`--with-gd`。要编译GD库，需要libpng和libjpeg。
- 对于Windows用户来说，只需在PHP的配置文件`php.ini`中，把配置项“`;extension=php_gd2.dll`”前的分号去掉即可。当正确配置`php.ini`中有关GD的项后，通过`phpinfo()`可以看到GD库被正确加载了，



## 19.1.2 主要图像处理函数介绍

- 目前GD2库的图像处理函数有上百个，限于篇幅不能一一讲述，本节将向读者介绍一些有代表性的图像处理函数。在下节，向读者通过实例向读者演示这些函数的用法。本节向读者介绍的函数如下所示。
- 函数 `resource imagecreatefromgif(string $filename)`，用来从给定的GIF文件或URL取出一图像，参数 `$filename` 是文件名或URL。该函数返回值是图像标识符，代表了从给定的文件名取得的图像。失败时，返回一空字符串，并且输出一条错误信息。
- 函数 `bool imagegif(resource $image [, string $filename])`，该函数从参数 `$image` 所代表的图像以参数 `$filename` 为文件名创建一个GIF图像。 `image` 参数是 `imagecreate()` 或 `imagecreatefromgif` 等函数的返回值。
- 函数 `resource imagecreatefrompng ( string $filename )`，该函数从PNG文件或URL取出一图像，参数 `$filename` 是文件名或URL。该函数返回值是图像标识符，如果执行失败，函数返回一个空字符串，并且输出一条错误信息。



## 19.1.2 主要图像处理函数介绍

- 函数 `bool imagepng(resource $image [, string $filename])`，该函数类似 `imagegif()`，将GD图像流（参数 `$image` 代表）以PNG格式输出到标准输出（通常为浏览器），或者如果用参数 `$filename` 给出了文件名，则将其输出到该文件。
- 函数 `resource imagecreate(int $x_size, int $y_size)`，新建一个基于调色版的图像，参数 `$x_size` 和 `$y_size` 代表了创建图像的宽和高，该函数返回所创建图像的标识符。
- 函数 `resource imagecreatetruecolor(int $x_size, int $y_size)`，该函数返回一个图像标识符，它代表了一幅大小为 `$x_size` 和 `$y_size` 的黑色图像。
- 函数 `int imagecolorallocate(resource $image, int $red, int $green, int $blue)`，参数 `$image` 是图片标识符，参数 `$red`、`$green`、`$blue` 分别代表色系中的红色、绿色和蓝色（RGB），这些参数的取值范围是0到255，或者十六进制的 `0x00` 到 `0xFF`，例如代码 `imagecolorallocate($img, 255, 0, 0)` 表示设置图像 `$img` 的颜色为红色。该函数的返回值代表了由给定的参数组成的颜色。



## 19.1.2 主要图像处理函数介绍

- 函数 `bool imagefill(resource $image, int $x, int $y, int $color)`，该函数在参数 `$image` 所指定图像的坐标 `$x` 和 `$y`（图像左上角为 `0, 0`）处用 `$color` 颜色执行区域填充，即与 `x, y` 点颜色相同且相邻的点都会被填充。
- 函数 `bool imageline(resource $image, int $x1, int $y1, int $x2, int $y2, int $color)`，该用参数 `$color` 所指定的颜色在参数 `$image` 所标识的图像中从坐标 `$x1, $y1` 到 `$x2, $y2`（图像左上角为 `0, 0`）画一条线段。
- 函数 `bool imagestring(resource $image, int $font, int $x, int $y, string $s, int $col)` 水平地显示一行字符串。该函数用参数 `$col` 所指定的颜色将字符串 `$s` 显示到参数 `$image` 所标识图像的 `$x, $y` 坐标处。
- 本章后续内容中还会涉及到一些本小节没有介绍到的函数，这些函数将在后续内容的实例中使用到时，再做详细介绍。



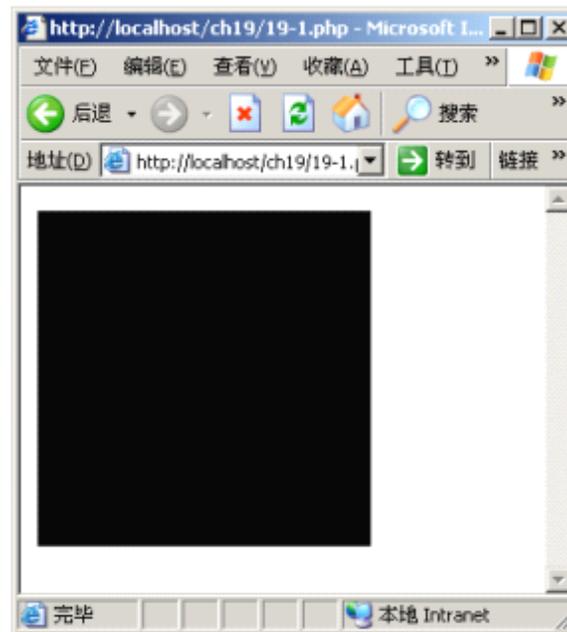
## 19.2 图像的建立

- 有了以上处理图像的PHP函数作为基础，本节读者可以尝试通过这些函数创建图像。使用PHP创建图像的步骤通常有以下4个步骤。
  - (1) 建立画布。
  - (2) 在画布上绘制形状或书写文本。
  - (3) 输出最终的图片。
  - (4) 清空绘图资源。



## 19.2.1 用PHP生成图像

- 下面的实例演示了如何是PHP函数创建一个图像，并将图像输出到页面。

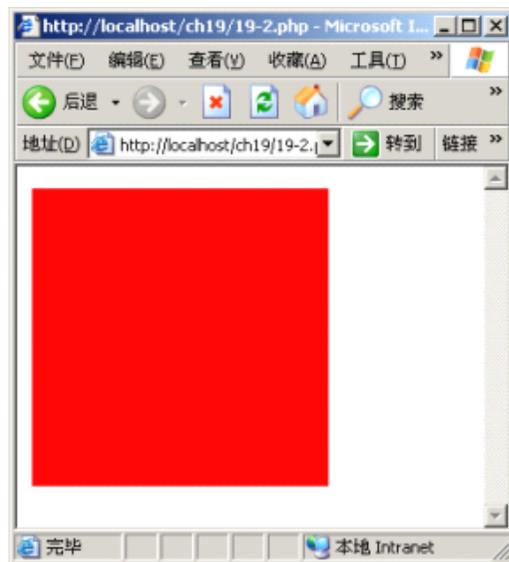


使用 PHP 的 GD 函数生成简单图像



## 19.2.2 设定图像颜色

- 函数 `imagecreatetruecolor()` 所返回的图像标识符，默认情况下代表一个黑色图像。可以通过函数 `imagecolorallocate()`，改变该表示符所代表图像的颜色。



设置图像颜色



## 19.2.3 在图像上绘制直线

- 本节将向读者介绍如何在图像上绘制形状或书写文字，也就是本节开头所属PHP图像处理的第2个步骤。本节将介绍在图像上绘制直线的函数。通过函数`imageline()`可以实现在图像中绘制直线，



在图像上绘制直线



## 19.2.4 在图像上输出文字

- 通过函数 `imagestring()` 可以在图像上书写一段文本，
- 函数 `imagefttext()` 的第1个参数是图像标识符，第2个参数是字体大小，第3个参数是文本显示角度，如果该参数为0，表示从左向右读的文本。第4、5个参数表示的第一个字符字体基线的坐标。第6个参数表示字体颜色。第7个参数表示显示图像文字时所采用的TrueType字体。最后一个参数是要显示的文本内容。



在图像上显示文字



在图像中显示中文



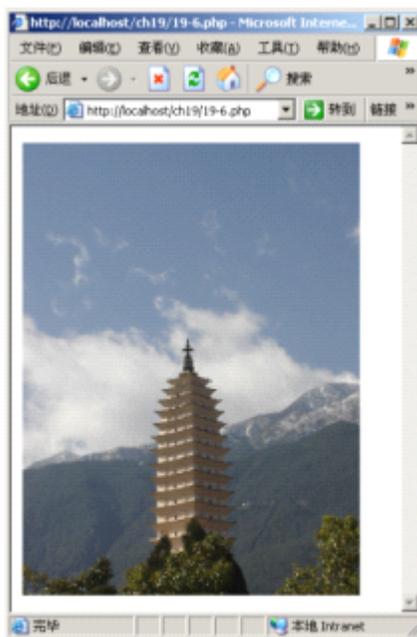
## 19.3 图片处理

- 上节简述了如何创建图像，并在图像上输出文字或线条。本节介绍对已有图片做处理，包括的内容有：如何打开现有图片、如何获取和该图片有关的信息、加工处理后输出图片、生成缩略图等。



## 19.3.1 打开已经存在的图片

- 通过19.1.2小节对一些函数的介绍，读者已经了解到通过函数imagecreatefrompng()可以打开一个已经存在的PNG格式的图片，



打开已存在的图片



## 19.3.2 获取图片的相关属性

- 对打开的图片可以获取该图片的有关属性，如图片大小、图片的宽和高等。



获取图像的宽、高属性



函数 getimagesize() 的用法



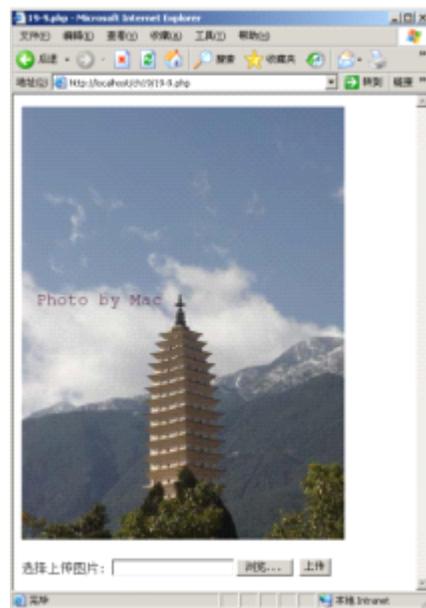
## 19.3.3 对图片加水印效果

- 对图片可以做各式各样的处理，以达到修饰图片的作用。本小节通过为图片添加水印效果为例，向读者介绍如何对图片做效果处理。
- 为图片添加水印通常有以下步骤。
- 获取要添加水印的图像的宽、高值。
- 确定图片大小是否满足水印文字大小。
- 确定水印效果在图片中的位置。
- 设定图像的混色模式。
- 生成水印效果。
- 释放资源。



## 19.3.4 生成已有图片的缩略图

- 对一张大图生成一个缩略图，也可以看作是图片效果处理的一部分。使用GD库的“拷贝并调整大小”的函数可以很容易实现生成缩略图，这个函数就是`imagecopyresized()`，



为图片添加水印效果



## 19.4 实例：生成带有底纹的数字验证码图片

- 简单地说，验证码就是网站为了防止用户利用网络“机器人”程序自动注册、登录、灌水发帖而采用的一种网络编程安全技术。
- 实质上，所谓的验证码，是通过网站编程实现的一项简单的功能，它就是将一串随机产生的数字或符号，生成一幅图片，图片里加上一些干扰象素（防止OCR），由用户肉眼识别其中的验证码信息，输入表单提交网站验证，验证成功后才能使用某项功能。图形验证码干扰色越多，自动机器人程序越难把字码认出来。通常用的干扰色都有：随机文字颜色、随机颜色底纹、随机文字位置、随机文字大小，随机背景花纹。



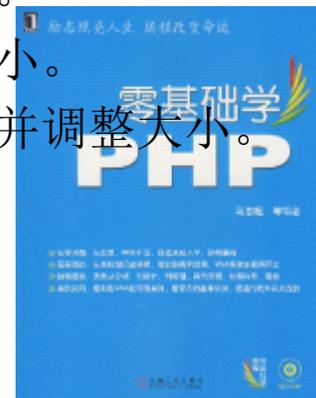
## 19.5 小结

- 本章向读者介绍了使用 PHP 的 GD 扩展库中的函数，完成生成图像以及图片的处理，首先介绍了 GD 库的安装及 GD 库中常用的函数，然后通过具体实例演示这些函数的用法。本节介绍的 GD 扩展库的主要函数如下所示。
- 函数 `imagecreatefromgif ()` 用来从指定的 GIF 文件取出图像。
- 函数 `imagegif ()` 用来创建一个 GIF 图像。
- 函数 `imagecreatefrompng ()` 用来从指定的 PNG 文件取出图像。
- 函数 `imagepng ()` 用来创建一个 PNG 图像。
- 函数 `imagecreate ()` 用来新建一个基于调色版的图像。



## 19.5 小结

- 函数`imagecreatetruecolor()` 返回一个黑色图像的标识符。
- 函数`imagecolorallocate()` 创建一个由RGB颜色指定的图像。
- 函数`imagefill()` 用指定的颜色填充图像。
- 函数`imageline()` 用指定的颜色画出一条线段。
- 函数`imagestring()` 在图像上显示一个字符串。
- 函数`imagefttext()` 用TrueType字体向图像写入文本。
- 函数`imageSX()/imageSY()` 分别用来取得图像宽度和高度。
- 函数`imageftbbox()` 用来取得使用TrueType字体的文本的范围。
- 函数`imagealphablending()` 用来设定图像的混色模式。
- 函数`imagecopyresized()` 用来拷贝部分图像并调整大小。
- 函数`imagecopyresampled()` 用来重采样拷贝部分图像并调整大小。



## 19.5 小结

- 除了介绍基本的图像处理函数之外，还介绍了一个不属于GD库的函数，它是获取图像属性的函数 `getimagesize()`。最后，本章介绍了大量处理图像的示例 PHP 程序，包括以下几项内容。
- 用PHP生成图像。
- 设定图像的颜色。
- 在图像上绘制直线。
- 在图线上显示英文和中文字符。
- 打开已存在的图片。
- 获取图片相关属性。
- 对图片添加水印效果。
- 生成图片缩略图。
- 生成带有底纹的数字验证码图片。

