python-PIL参考手册

一，使用img类

1.format 属性表示图像的原始格式。如果图像不是从文件中读取的，则它被设置成None。

2.size 属性是一个2元组，表示图像的宽度和高度（以像素为单位）。

3. mode 属性定义图像的色彩通道的数量与名字，同时也包括像素的类型和颜色深度信息。通常来说，灰度图像的mode是"L" (luminance)，真彩色图像的mode是"RGB" ，而用来打印的图像的mode是"CMYK"。

注意：如果文件不能打开，则会抛出一个IOError异常。

4，img显示

show 的标准实现不是很高效，因为它先将图像保存成一个临时文件，然后调用xv程序来显示图像。如果你没有安装xv，他甚至不能工作。然而如果他可用，他将非常方便。

5,读写图像

从磁盘读取文件，使用open函数，库自动根据内容确定格式；
保存文件，使用save函数，要指定格式，否则按扩展名储存。

6，裁剪，粘贴合并图像

使用crop方法

7，Python Imaging Library 还允许对一幅多通道图像（比如RGB图像）的单个通道进行操作。split方法能够创建一组新的图像，每一幅都是原来多通道图像的一个通道。merge函数以一个模式和一组图像的元组为参数，把这些 图像组成 一幅 新图像。 下面的例子实现交换一幅RGB图像的三个通道：

分离与合并通道

切换行号显示

```
r, g, b = im.split
im = Image.merge("RGB", (b, g, r))
```

二， 几何变化

1，resize缩放 带一个tuple类型的参数来表示新图片大小。

2， rotate旋转 带一个逆时针角度值作为参数

3，transpose图像，更通用的方法transform.

三， 颜色变化

convert函数

注意：库支持在所有支持的颜色模式和"L"以及"RGB"之间的直接转换。其他颜色模式之间的转换要借助于中间图像 模式（通常是"RGB" 模式）。

四， 图像增强

1，滤波器

ImageFilter 模块中包含一些预定义的增强滤波器，用filter 方法来使用滤波器。

eg:import ImageFilter

```
out = im.filter(ImageFilter.DETAIL)
```

2，点操作

point方法可以对图像的像素点进行变化（比如对比度变换）。

eg:out = im.point(lambda i: i * 1.2)

可结合 paste 方法使用

五，图像序列（动画格式）

1，使用seek和tell方法在不同帧之间移动。

注意，当前版本库的绝大多数驱动只允许你移动到下一帧(如上面例子所示)。如果要回到文件的开头，你可能必须重新打开它。

2，Postscript格式打印。

3，控制解码器。允许在读取图像的时候对图像进行操作。常被用来制作缩略图。

六，处理光栅图像，即方形的像素数据。

1，通道。

一幅图像可以有一个或者多个通道的数据构成。Python Imaging Library允许在一个图像中存储多个通道，只要这些通道的大小和颜色深度都是一样的。要获取图像的通道数目和通道名称，可以使用[image.htm#image- getbands-method getbands] 方法。

2，大小

通过图像的[image.htm#image-size-attribute size]属性可以读取图像的大小信息。大小信息由一个包含水平 和垂直像素数的二元组表示。

3，坐标系统

Python Imaging Library 使用笛卡尔像素坐标系统，原点（0,0)在图像的左上角。

注意：坐标值对应像素的左上角，像素(0, 0)实际中心位于(0.5, 0.5)。 坐标通常以元组(x, y)

的形式传递给库。矩形则表示成

4元组的形式，左上角为第一个。比如，覆盖整个800x600像素的矩形表示为(0, 0, 800, 600)。

4，调色板

模式 ("P")使用一个彩色调色板来定义每个像素的真实颜色。

5，在图像中添加辅助信息。

6，滤波器。

二，模块