



# Python语言基础与应用

高级扩展模块 / 图像处理库

陈斌 北京大学 [gischen@pku.edu.cn](mailto:gischen@pku.edu.cn)

# 图像处理库

- › **Pillow库**
- › **图像处理步骤**
- › **PIL图像操作**
- › **小程序：PIL生成验证码**



# Pillow库

## › Python里的图像处理库

PIL: Python Image Library

## › 功能强大，可以对图像做各种处理

缩放、裁剪、旋转、滤镜、文字、调色板等等



# 图像处理步骤

## › 打开图像

`image.open(<路径+图像名+文件格式>)`

Pillow库能自动根据文件内容确定格式

若图片在程序目录下，则不需要附带路径，直接将图像名+文件格式作为参数

## › 处理图像

image模块中提供了大量处理图像的方法

## › 存取或显示图像

`im.show()`

`im.save(<文件名>)`







# PIL图像操作：缩略图












## › thumbnail函数

`thumbnail(size, Image.ANTIALIAS)`

参数`size`为一个元组，指定生成缩略图的大小

直接对内存中的原图进行了修改，但是修改完后的图片需要保存，处理后的图片不会被拉伸

新加卷 (D:) > 第一行代码 > 缩略图			
<input type="checkbox"/> 名称	修改日期	类型	大小
 创建缩略图	2018/5/16 11:55	PY 文件	
 海龟画图4	2017/12/1 14:52	JPG 文件	
 海龟画图3	2017/12/1 14:45	JPG 文件	
 海龟作图2	2017/12/1 14:44	JPG 文件	
 海龟画图1	2017/12/1 14:44	JPG 文件	
 test	2017/11/20 14:57	JPG 文件	

新加卷 (D:) > 第一行代码 > 缩略图			
<input type="checkbox"/> 名称	修改日期	类型	大小
 test.thumbnail	2018/5/16 12:07	THUMBNAI...	3 KB
 海龟画图1.thu...	2018/5/16 12:07	THUMBNAI...	3 KB
 海龟画图3.thu...	2018/5/16 12:07	THUMBNAI...	3 KB
 海龟画图4.thu...	2018/5/16 12:07	THUMBNAI...	3 KB
 海龟作图2.thu...	2018/5/16 12:07	THUMBNAI...	3 KB
 创建缩略图	2018/5/16 12:06	PY 文件	1 KB
 海龟画图4	2017/12/1 14:52	JPG 文件	46 KB
 海龟画图3	2017/12/1 14:45	JPG 文件	35 KB
 海龟作图2	2017/12/1 14:44	JPG 文件	204 KB
 海龟画图1	2017/12/1 14:44	JPG 文件	213 KB
 test	2017/11/20 14:57	JPG 文件	52 KB

# PIL图像操作：模糊效果

```
from PIL import Image, ImageFilter

# 打开一个jpg图像文件，注意是当前路径：
im = Image.open('test.jpg')
# 应用模糊滤镜：
im2 = im.filter(ImageFilter.BLUR)
im2.save('blur.jpg', 'jpeg')
```





# PIL图像操作：添加文字

```
from PIL import Image, ImageDraw, ImageFont
#打开程序目录下的图片cat
img = Image.open('cat.jpg')
#设置待添加文字大小为200, 字体为宋体
font = ImageFont.truetype('simsum.ttc', 100)
#在img上创建可绘图对象draw
draw = ImageDraw.Draw(img)
#添加红色文字“可爱的小猫”
draw.text((100, 10), '可爱的小猫', (255, 0, 0), font=font)
#保存图片
img.save('cat1.jpg', 'jpeg')
```



# 小程序：PIL生成验证码

```
from PIL import Image, ImageDraw, ImageFont, ImageFilter
import random

# 随机字母:
def rndChar():
    return chr(random.randint(65, 90))

# 随机颜色1:
def rndColor():
    return (random.randint(64, 255), \
            random.randint(64, 255), \
            random.randint(64, 255))

# 随机颜色2:
def rndColor2():
    return (random.randint(32, 127), \
            random.randint(32, 127), \
            random.randint(32, 127))
```



# 小程序：PIL生成验证码

```
# 240 x 60:
width = 60 * 4
height = 60
image = Image.new('RGB', (width, height), (255, 255, 255))
# 创建Font对象:
font = ImageFont.truetype('Arial.ttf', 36)
# 创建Draw对象:
draw = ImageDraw.Draw(image)
# 填充每个像素:
for x in range(width):
    for y in range(height):
        draw.point((x, y), fill=rndColor())
# 输出文字:
for t in range(4):
    draw.text((60 * t + 10, 10), rndChar(), font=font, fill=rndColor2())
# 模糊:
image = image.filter(ImageFilter.BLUR)
image.save('code.jpg', 'jpeg')
```

