

Python程序设计

陈远祥

chenyxmail@gmail.com

北京邮电大学 电子工程学院



Python程序设计

上周主要内容

- 组合数据类型及其操作
- 正则表达式和模式匹配

文本词频统计

✓ 三国人物

```
三国人物1.py - F:\北邮课程\Python程序设计\备课\第五周\三国人物1.py (3.6.4)
File Edit Format Run Options Window Help

import jieba
f=open("三国演义.txt", "rb")
txt=f.read()
words = jieba.lcut(txt)
counts = {}
for word in words:
    if len(word) == 1: #排除单个字符的分词结果
        continue
    else:
        counts[word] = counts.get(word, 0) + 1

items = list(counts.items())
items.sort(key=lambda items:items[1], reverse=True)
for i in range(20):
    word, count = items[i]
    print (" {0:<10} {1:>5}".format(word, count))
```

利用正则表达式从网页提取图片

图片.py - F:\北邮课程\Python程序设计\备课\第五周\抓取图片\图片.py (3.6.4)

File Edit Format Run Options Window Help

```
#抓取图片
import re
import urllib #urllib库，操作URL的功能
#获取网页信息
url="https://bj.lianjia.com/zufang/"
page = urllib.request.urlopen(url)#抓取网页的源代码
html = page.read()#bytes类型
html=html.decode('utf-8')#str类型
#获取图片，正则表达式
reg=r'data-img="(.*?\.(jpg))" alt'#源代码格式图片
imgre=re.compile(reg)
imglist = re.findall(imgre,html)
#抓取页面图片并保存到本地
x=0
for img in imglist:
    urllib.request.urlretrieve(img, '%s.jpg' % x)#直接将远程数据下载到本地
    x+=1
```

正则表达式

■ 为什么需要正则表达式？

- ✓ 比如123456@qq.com，我们一看就是邮箱，可是计算机不认识
- ✓ 用计算机认识的语言，制定好规则，告诉它符合这个规则的就是个邮箱，这样计算机就能帮我们找到对应的东西
- ✓ 正则就是用来设置规则，来完成我们需求的一些操作的，比如登录验证，搜索指定的东西

正则表达式作用

- 1. **test**: 对字符串进行判断
 - ✓ 判断是否是数字
 - ✓ 判断密码是否是字母、数字组合

正则表达式作用

- 2. `search`: 在字符串搜索符合正则的内容，搜索到就返回出现的位置
 - ✓ 在字符串中找字母b，且不区分大小写

正则表达式作用

- 3. `match`: 在字符串中搜索符合规则的内容，搜索成功就返回内容，格式为数组，失败就返回 `none`
 - ✓ 找出指定格式的所有数字

正则表达式作用

- 4. **replace**: 查找符合正则的字符串，就替换成对应的字符串。返回替换后的内容
 - ✓ 敏感词过滤

各种各样的正则表达式

 正则表达式在线测试

生成代码

我的邮箱地址是chenyxmail@butp.edu.cn

正则表达式:

☒ 全局搜索 ☐ 忽略大小写

测试匹配

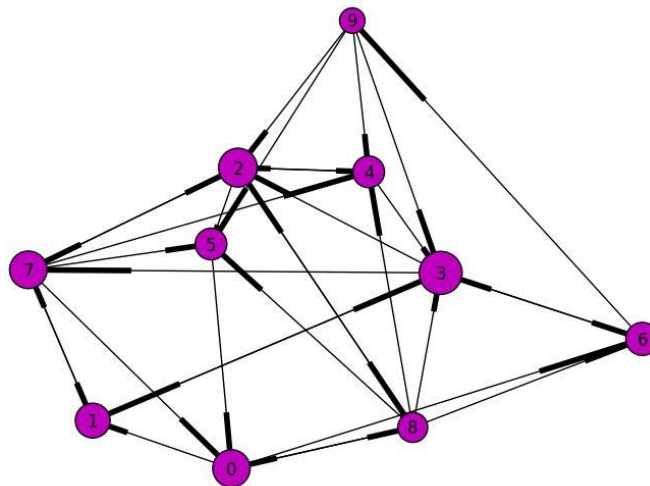
共找到 1 处匹配:
chenyxmail@butp.edu.cn

<https://c.runoob.com/front-end/854>

课后作业

- 1、从网页（链家租房）获取房源信息
- 2、提取小区名字，房屋面积和价格
- 3、按价格或者面积对房源进行排序

Python计算生态和模块化编程



计算生态和模块化编程

- Python语言有9万多个第三方库，形成了庞大的计算生态，模块编程思想是Python语言最大的价值
- 近20年的开源运动产生了深植于各信息技术领域的大量可重用资源，直接且有力的支撑了信息技术超越其他技术领域的发展速度，形成了“计算生态”
- 产业界广泛利用可重用资源快速构建应用已经是主流产品开发方式

计算生态和模块化编程

■ Python的计算生态

- ✓ Python语言从诞生之初致力于开源开放，建立了全球最大的编程计算生态
- ✓ Python官方网站提供了第三方库索引功能

<https://pypi.python.org/pypi>

计算生态和模块化编程

- 该页面列出了Python语言9万多个第三方库的基本信息，这些函数库覆盖信息领域技术所有技术方向
- Python语言的函数库并非都采用Python语言编写

计算生态和模块化编程

■ 计算生态发展历程：

- ✓ 30年前，编写程序仅能调用官方提供的API功能
- ✓ 20年前，开源运动的兴起和蓬勃发展，一批开源项目诞生
- ✓ 10年前，开源运动深入开展，专业人士开始大量贡献各领域最优秀的研究和开发成果，并通过开源库形式发布出来
- ✓ 那今天呢？编程领域形成了庞大的计算生态，需要一种编程语言或方式能够将不同语言、不同特点、不同使用方式的代码统一起来

计算生态和模块化编程

- 由于Python有非常简单灵活的编程方式，很多采用C、C++等语言编写的专业库可以经过简单的接口封装供Python语言程序调用。这样的粘性功能使得Python语言成为了各类编程语言之间的接口，俗称Python语言为“胶水语言”

计算生态和模块化编程

- Python 第三方程序包括库（library）、模块（module）、类（class）和程序包（Package）等多种命名
- 统一将这些可重用代码统称为“库”

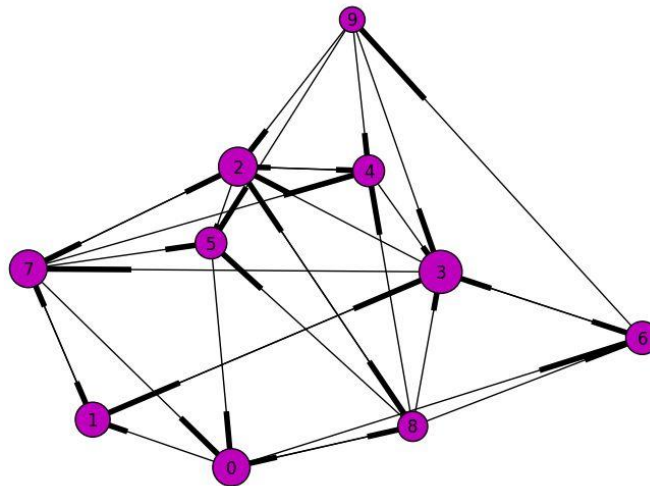
计算生态和模块化编程

- Python内置的库称为标准库，其他库称为第三方库
- 库编程：在计算生态思想指导下，编写程序的起点不再是探究每个具体算法的逻辑功能和设计，而是尽可能利用第三方库进行代码复用，探究运用库的系统方法

计算生态和模块化编程

- 这种像搭积木一样的编程方式，称为“模块编程”
- 每个模块可能是标准库、第三方库、用户编写的其他程序或对程序运行有帮助的资源等

Python 第三方库的安装



第三方库的安装

- 第三方库需要安装后才能使用。由于Python语言经历了版本更迭过程，而且，第三方库由全球开发者分布式维护，缺少统一的集中管理，因此，Python的第三方库曾经一度制约了Python语言的普及和发展。随着官方pip工具的应用，Python第三方库的安装变得十分容易

第三方库的安装

- Python第三方库依照安装方式灵活性和难易程度有三个方法
- 三个方法：`pip`工具安装、自定义安装和文件安装

第三方库的安装

■ pip 工具安装

- ✓ 最常用且最高效的Python第三方库安装方式是采用pip工具安装
- ✓ pip是Python官方提供并维护的在线第三方库安装工具
- ✓ 对于同时安装Python2和Python3环境的系统, 建议采用pip3命令专门为Python3版本安装第三方库

第三方库的安装

■ pip 工具安装

- ✓ pip是Python内置命令，需要通过命令行执行，执行pip-h命令将列出pip常用的子命令，注意，不要在IDLE环境下运行pip程序
- ✓ 启动anaconda命令窗口:anaconda prompt
- ✓ pip -h

第三方库的安装

■ pip 工具安装

- ✓ pip 支持安装（install）、下载（download）、卸载（uninstall）、列表（list）、查看（list）、查找（search）等一系列安装和维护子命令
- ✓ 安装一个库的命令格式如下，例如：
`pip install <拟安装库名>`

第三方库的安装

■ pip 工具安装

- ✓ 可以通过list子命令列出当前系统中已经安装的第三方库，例如：
- ✓ `pip list`

第三方库的安装

- `pip`是Python第三方库最主要的安装方式，可以安装超过90%以上的第三方库。然而，由于一些历史、技术和政策等原因，还有一些第三方库无法暂时用`pip`安装，此时，需要其他的安装方法

第三方库的安装

- pip工具与操作系统也有关系，在MacOSX和Linux等操作系统中，pip工具几乎可以安装任何Python第三方库，在Windows操作系统中，有一些第三方库仍然需要用其他方式尝试安装

第三方库的安装

■ 自定义安装

- ✓ 自定义安装指按照第三方库提供的步骤和方式安装。第三方库都有主页用于维护库的代码和文档。以numpy为例，开发者维护的官方主页是：<http://www.numpy.org/>
- ✓ 浏览该网页找到下载链接，如下：
- ✓ <http://www.scipy.org/scipylib/download.html>
- ✓ 进而根据指示步骤安装。自定义安装一般适合在pip中尚无登记或安装失败的

第三方库的安装

■ 文件安装

- ✓ 由于Python某些第三方库仅提供源代码，通过pip下载文件后无法在Windows系统编译安装，会导致第三方库安装失败。在Windows平台下遇到的无法安装第三方库的问题大多属于这类

第三方库的安装

■ 文件安装

- ✓ 为了解决这类第三方库安装问题，美国加州大学尔湾分校提供了一个页面，帮助Python用户获得Windows可直接安装的第三方库文件，
- ✓ 链接地址如下：

<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

第三方库的安装

■ 文件安装

- ✓ 该地址列出了一批在pip安装中可能出现问题的第三方库
- ✓ 这里以sklearn为例说明，首先在上述页面中找到sklearn库对应的内容

第三方库的安装

■ 文件安装

✓ 选择其中的.whl文件下载

`Py-earth`, an implementation of Jerome Friedman's multivariate adaptive regression splines.

`sklearn contrib py earth-0.1.0-cp27-cp27m-win32.whl`

`sklearn contrib py earth-0.1.0-cp27-cp27m-win amd64.whl`

`sklearn contrib py earth-0.1.0-cp34-cp34m-win32.whl`

`sklearn contrib py earth-0.1.0-cp34-cp34m-win amd64.whl`

`sklearn contrib py earth-0.1.0-cp35-cp35m-win32.whl`

`sklearn contrib py earth-0.1.0-cp35-cp35m-win amd64.whl`

`sklearn contrib py earth-0.1.0-cp36-cp36m-win32.whl`

`sklearn contrib py earth-0.1.0-cp36-cp36m-win amd64.whl`

`sklearn contrib py earth-0.1.0-cp37-cp37m-win32.whl`

`sklearn contrib py earth-0.1.0-cp37-cp37m-win amd64.whl`

第三方库的安装

■ 文件安装

✓ 然后，采用pip命令安装该文件

```
(base) C:\Users\chenfox>pip install sklearn_contrib_py_earth-0.1.0-cp36-cp36m-win_amd64.whl
Processing c:\users\chenfox\sklearn_contrib_py_earth-0.1.0-cp36-cp36m-win_amd64.whl
Requirement already satisfied: scikit-learn>=0.16 in c:\users\chenfox\anaconda3\lib\site-packages (from sklearn-contrib-py-earth==0.1.0)
Requirement already satisfied: scipy>=0.16 in c:\users\chenfox\anaconda3\lib\site-packages (from sklearn-contrib-py-earth==0.1.0)
Requirement already satisfied: six in c:\users\chenfox\anaconda3\lib\site-packages (from sklearn-contrib-py-earth==0.1.0)
Requirement already satisfied: numpy>=1.8.2 in c:\users\chenfox\anaconda3\lib\site-packages (from scipy>=0.16->sklearn-contrib-py-earth==0.1.0)
Installing collected packages: sklearn-contrib-py-earth
Successfully installed sklearn-contrib-py-earth-0.1.0

(base) C:\Users\chenfox>
```

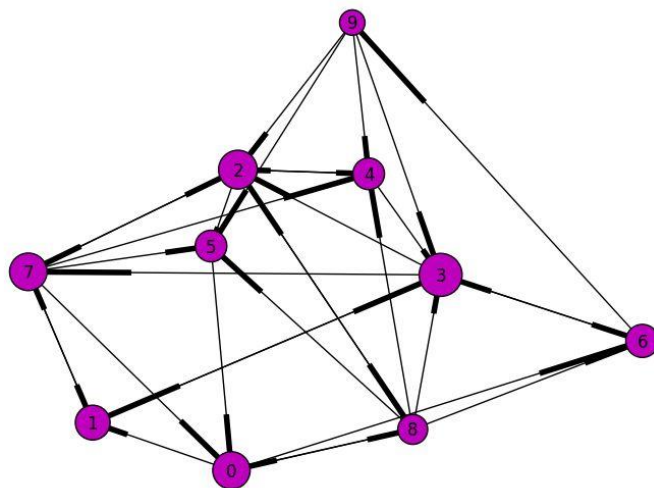
第三方库的安装

- 对于上述三种安装方式，一般优先选择采用pip工具安装，如果安装失败，则选择自定义安装或者文件安装（Windows平台）。另外，如果需要在没有网络条件下安装Python第三方库，请直接采用文件安装方式。其中，.whl文件可以通过pipdownload指令在有网络条件的情况下获得

第三方库的安装

- .whl 是 Python 库的一种打包格式，用于通过 pip 进行安装，相当于 Python 库的安装包文件。 .whl 文件本质上是一个压缩格式文件，可以通过变化扩展名为 .zip 查看其中内容。 .whl 格式用于替代 Python 早期的 eggs 格式，是 Python 打包格式的事实标准

文件



文件

- 存储在外部介质上的数据或信息的集合
 - ✓ 程序中的源程序
 - ✓ 数据中保存着数据
 - ✓ 图像中的像素数据
 - ✓ ...
- 有序的数据序列

文件

- 编码：信息从一种形式转换为另一种形式的过程
 - ✓ ASCII 码
 - ✓ Unicode
 - ✓ UTF-8
 - ✓ ...

文件

■ ASCII码

- ✓ ASCII (American Standard Code for Information Interchange), 是一种单字节的编码。计算机世界里一开始只有英文, 而单字节可以表示256个不同的字符, 可以表示所有的英文字符和许多的控制符号

文件

■ Unicode

- ✓ 跨语言、跨平台进行文本转换和处理
- ✓ 对每种语言中字符设定统一且唯一的二进制编码
- ✓ 每个字符两个或4个字节长
- ✓ 每个英文字母前都必然有二到三个字节是0, 这对于存储来说是极大的浪费

文件

■ UTF-8编码

- ✓ 可变长度的Unicode的实现方式
- ✓ 使用1~4个字节表示一个符号，根据不同的符号而变化字节长度

文件

- 文件包括两种类型：文本文件和二进制文件
- 二进制文件：直接由比特0和比特1组成，没有统一字符编码，文件内部数据的组织格式与文件用途有关
- 文本文件：以ASCII码方式存储的文件
- 二进制文件和文本文件最主要的区别在于是否有统一的字符编码

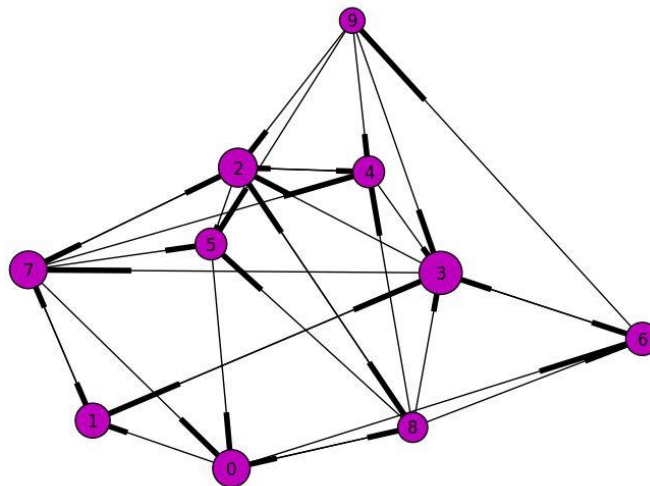
文件

- 无论文件创建为文本文件或者二进制文件，都可以用“文本文件方式”和“二进制文件方式”打开，打开后的操作不同
- 采用文本方式读入文件，文件经过编码形成字符串，打印出有含义的字符；采用二进制方式打开文件，文件被解析为字节（byte）流。文本文件由于存在编码，字符串中的一个字符由2个字节表示
- ✓ #文件

文件

```
1 textFile=open("文件.txt","rt")#t表示文本文件方式
2 print(textFile.readline())
3 textFile.close()
4 binFile=open("文件.txt","rb")#r表示二进制文件方式
5 print(binFile.readline())
6 binFile.close()
```

文件的操作



文件的操作

■ 打开文件

- ✓ 建立磁盘上的文件与程序中的对象相关联
- ✓ 通过相关的文件对象获得

■ 文件操作

- ✓ 读取
- ✓ 写入
- ✓ 定位
- ✓ 其他：追加、计算等

■ 关闭文件

- ✓ 切断文件与程序的联系
- ✓ 写入磁盘，并释放文件缓冲区

文件的操作

■ 打开文件：

Python通过解释器内置的`open()`函数打开一个文件，并实现该文件与一个程序变量的关联，`open()`函数格式如下：

`<变量名>=open(<文件名>,<打开模式>)`

`open()`函数有两个参数：文件名和打开模式。文件名可以是文件的实际名字，也可以是包含完整路径的名字

文件的操作

■ open () 函数提供7种基本的打开模式

打开模式	含义
'r'	只读模式，如果文件不存在，返回异常FileNotFoundError，默认值
'w'	覆盖写模式，文件不存在则创建，存在则完全覆盖源文件
'x'	创建写模式，文件不存在则创建，存在则返回异常FileExistsError
'a'	追加写模式，文件不存在则创建，存在则在原文件最后追加内容
'b'	二进制文件模式
't'	文本文件模式，默认值
'+'	与r/w/x/a一同使用，在原功能基础上增加同时读写功能

文件的操作

- 文件的读写：根据打开方式不同可以对文件进行相应的读写操作，Python提供4个常用的文件内容读取方法

方法	含义
<code><file>.readall()</code>	读入整个文件内容，返回一个字符串或字节流*
<code><file>.read(size=-1)</code>	从文件中读入整个文件内容，如果给出参数，读入前size长度的字符串或字节流
<code><file>.readline(size = -1)</code>	从文件中读入一行内容，如果给出参数，读入该行前size长度的字符串或字节流
<code><file>.readlines(hint=-1)</code>	从文件中读入所有行，以每行为元素形成一个列表，如果给出参数，读入hint行

文件的操作

■ 文本文件逐行打印

✓ #文件2

如果程序需要逐行处理文件内容，建议采用上述代码格式：

```
fo=open(fname, "r")  
for line in fo:  
    #处理一行数据  
fo.close()
```

文件的操作

```
1 #fname=input("请输入要打开的文件:")
2 #fo=open(fname,"rt")
3 #for line in fo.readlines():
4 #     print(line)
5 #fo.close()
6 ##
7 fname=input("请输入要打开的文件:")
8 fo=open(fname,"rt")
9 for line in fo:
10     print(line)
11 fo.close()
```

文件的操作

- 文件的读写：Python提供3个与文件内容写入有关的方法

方法	含义
<code><file>.write(s)</code>	向文件写入一个字符串或字节流
<code><file>.writelines(lines)</code>	将一个元素为字符串的列表写入文件
<code><file>.seek(offset)</code>	改变当前文件操作指针的位置，offset的值： 0：文件开头； 1：当前位置； 2：文件结尾

✓ #文件3

文件的操作

```
1 fname=input("请输入要写入的文件:")
2 fo=open(fname,"w+")
3 ls=["唐诗","宋词","元曲"]
4 fo.writelines(ls)
5 for line in fo:
6     print(line)
7 fo.close()#有无close的区别
```

文件的操作

- 编写程序根据文件data.txt中的数据，使用turtle库来动态绘制图形路径

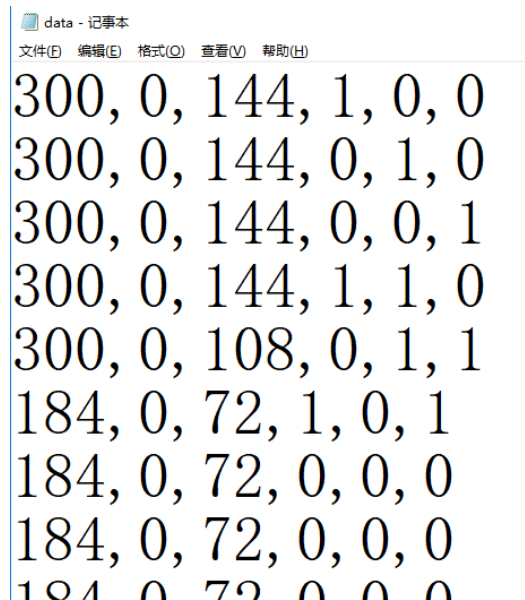
- 前进300个像素

- 0: 向左转

- 1: 向右转

- 转动多少角度

- 颜色: r, g, b



```
data - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
300, 0, 144, 1, 0, 0
300, 0, 144, 0, 1, 0
300, 0, 144, 0, 0, 1
300, 0, 144, 1, 1, 0
300, 0, 108, 0, 1, 1
184, 0, 72, 1, 0, 1
184, 0, 72, 0, 0, 0
184, 0, 72, 0, 0, 0
104 0 72 0 0 0
```

- ✓ #turtle1

文件的操作

- 编写程序将电话簿TeleAddressBook.txt和电子邮件EmailAddressBook.txt 合并为一个完整的AddressBook.txt

teleAddressBook - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

姓名	电话号码
小明	1520098751
小花	1520098752
小强	1520098753
张三	1520098754
李四	1520098755

EmailAddressBook - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

姓名	邮箱
小明	1520098751@qq.com
小张	1520098689@qq.com
小强	1520098753@qq.com
张三	1520098754@qq.com

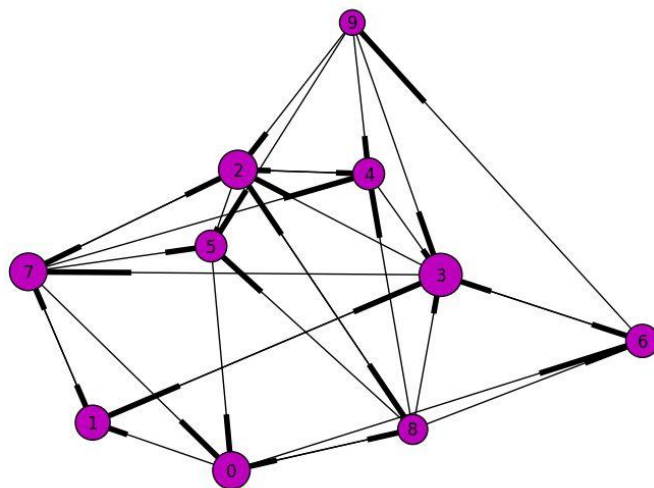
addressbook - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

姓名	电话	邮箱
小明	1520098751	1520098751@qq.com
小花	1520098752	----
小强	1520098753	1520098753@qq.com
张三	1520098754	1520098754@qq.com
李四	1520098755	----
小张	----	1520098689@qq.com

✓ #addressbook

PIL库的使用



PIL库的使用

- PIL库支持图像存储、显示和处理，它能够处理几乎所有图片格式，可以完成对图像的缩放、剪裁、叠加以及向图像添加线条、图像和文字等操作
- PIL (Python Image Library) 库是Python语言的第三方库，需要通过pip工具安装
`pip install pillow`

PIL库的使用

- PIL库可以完成图像归档和图像处理两方面功能需求：
 - ✓ 图像归档：对图像进行批处理、生成图像预览、图像格式转换等
 - ✓ 图像处理：图像基本处理、像素处理、颜色处理等

PIL库的使用

- 在PIL中，任何一个图像文件都可以用Image对象表示Image类的图像读取和创建方法

方法	描述
<code>Image.open(filename)</code>	根据参数加载图像文件
<code>Image.new(mode, size, color)</code>	根据给定参数创建一个新的图像
<code>Image.open(StringIO.StringIO(buffer))</code>	从字符串中获取图像
<code>Image.frombytes(mode, size, data)</code>	根据像素点data创建图像
<code>Image.verify()</code>	对图像文件完整性进行检查，返回异常

PIL库的使用

■ Image类有4个处理图片的常用属性

属性	描述
Image.format	标识图像格式或来源，如果图像不是从文件读取，值是None
Image.mode	图像的色彩模式，"L"灰度图像、"RGB"真彩色图像、"CMYK"出版图像
Image.size	图像宽度和高度，单位是像素（px），返回值是二元元组（tuple）
Image.palette	调色板属性，返回一个ImagePalette类型

■ Image类的图像转换和保存方法如表所示

方法	描述
Image.save(filename, format)	将图像保存为filename文件名，format是图片格式
Image.convert(mode)	使用不同的参数，转换图像为新的模式
Image.thumbnail(size)	创建图像的缩略图，size是缩略图尺寸的二元元组

PIL库的使用

- 生成"birdnest.jpg"图像的缩略图，其中（128, 128）是缩略图的尺寸
 - ✓ #缩略图

```
1 from PIL import Image
2 im=Image.open("birdnest.jpg")
3 im.thumbnail((128,128))
4 im.save("birdnestTN", "JPEG")
5
6
```

PIL库的使用

- Image类能够对每个像素点或者一幅RGB图像的每个通道单独进行操作, `split()` 方法能够将RGB图像各颜色通道提取出来, `merge()` 方法能够将各独立通道再合成一幅新的图像

方法	描述
<code>Image.point(func)</code>	根据函数func功能对每个元素进行运算, 返回图像副本
<code>Image.split()</code>	提取RGB图像的每个颜色通道, 返回图像副本
<code>Image.merge(mode, bands)</code>	合并通道, 采用mode色彩, bands是新色的色彩通道
<code>Image.blend(im1, im2, alpha)</code>	将两幅图片im1和im2按照如下公式插值后生成新的图像: $im1 * (1.0 - alpha) + im2 * alpha$

- ✓ #颜色变换, 通道颜色变换

PIL库的使用

```
1 from PIL import Image
2 im=Image.open('birdnest.jpg')
3 r,g,b=im.split()
4 om=Image.merge("RGB",(b,g,r))
5 om.save('birdnestBGR.jpg')
6
```

```
1 im=Image.open('birdnest.jpg')#打开鸟巢文件
2 r,g,b=im.split()#获得RGB通道数据
3 newr=g.point(lambda i:i*0.8)#将r通道颜色值变为原来的0.8倍
4 newg=g.point(lambda i:i<200)#选择g通道值低于200的像素点
5 newb=b.point(lambda i:i)
6 om=Image.merge(im.mode,(newr,newg,b))#将3个通道合形成新图像
7 om.save('birdnestMerge.jpg')#输出图片
8
```

PIL库的使用

- PIL库的 `ImageFilter` 类和 `ImageEnhance` 类提供了过滤图像和增强图像的方法，共10种

方法表示	描述
<code>ImageFilter.BLUR</code>	图像的模糊效果
<code>ImageFilter.CONTOUR</code>	图像的轮廓效果
<code>ImageFilter.DETAIL</code>	图像的细节效果
<code>ImageFilter.EDGE_ENHANCE</code>	图像的边界加强效果
<code>ImageFilter.EDGE_ENHANCE_MORE</code>	图像的阈值边界加强效果
<code>ImageFilter.EMBOSS</code>	图像的浮雕效果
<code>ImageFilter.FIND_EDGES</code>	图像的边界效果
<code>ImageFilter.SMOOTH</code>	图像的平滑效果
<code>ImageFilter.SMOOTH_MORE</code>	图像的阈值平滑效果
<code>ImageFilter.SHARPEN</code>	图像的锐化效果

- ✓ #图像的轮廓获取

PIL库的使用

```
1 from PIL import Image
2 from PIL import ImageFilter
3 im=Image.open('birdnest.jpg')
4 om=im.filter(ImageFilter.CONTOUR)
5 om.save('birdnestContour.jpg')
6
7
```

PIL库的使用

- **ImageEnhance** 类提供了更高级的图像增强需求，它提供调整色彩度、亮度、对比度、锐化等功能

方法	描述
<code>ImageEnhance.enhance(factor)</code>	对选择属性的数值增强factor倍
<code>ImageEnhance.Color(im)</code>	调整图像的颜色平衡
<code>ImageEnhance.Contrast(im)</code>	调整图像的对比度
<code>ImageEnhance.Brightness(im)</code>	调整图像的亮度
<code>ImageEnhance.Sharpness(im)</code>	调整图像的锐度

✓ #增强

PIL库的使用

```
1|from PIL import Image
2|from PIL import ImageEnhance
3|im=Image.open('birdnest.jpg')
4|om=ImageEnhance.Contrast(im)
5|om.enhance(100).save('birdnestEnContrast.jpg')
6|
7|
```

谢谢