a = [1,2,3]

b = [a,b,c]

zip(a,b)

[(1, a), (2, b), (3, c)]

或：

list(zip(a,b))

[(1, 4), (2, 5), (3, 6)]

>>>def func():

>>> global x使用全局变量

>>> x = 1

>>>

>>>func()

>>>print x

1

成员操作符：in ,not in

装饰器的作用："@名字"

自身加1：sum += 1

>>> a = [1, 2, 3, 4, 5]

>>> a + [6, 7, 8]

[1, 2, 3, 4, 5, 6, 7, 8]

>>> a - b a和b的差集

>>> a | b a和b的并集

str = '12345678'print str[0:1] #截取

str = 'a,hello'print str.find('hello') #在字符串str里查找字符串

str(obj) 转为字符串

abs(num) 求绝对值

type(obj) 返回对象类型

range([[start,]stop[,step]]) 返回整型列表

pass 仅用作占位，实际什么也不做

len() 返回长度

S.lower() 小写

S.upper() 大写

strip() 去掉左右两边空格

lstrip() 去掉左边空格

rstrip() 去掉右边空格

import random

random.randint(0,99) 随机整数

random.randrange(0, 101, 2) 随机偶数

random.uniform(1, 10) 随机浮点数

random.choice('abcdf#%^\*f') 随机字符

random.sample('abcdefj',3) 选取特定数量的字符

shoplist[0:4] 返回从位置0开始，包括位置1，2，3

nameinfo['a1']='yangrong' 增加字典元素，若有则覆盖

for (k,v) in aDict.items(): 遍历

print '%s:%s' %(k, v)

dict.has\_key('ob4') 判断是否存在主键

dict.items() 列出所有项目

del dict['xiaohei'] 删除xiaohei键值

dic = {'name': 'xiaoyi', 'age': 25, 'gender': 'male'}

获得所有键 dic.keys()

获得所有值 dic.values()

清空字典 dic.clear()

字典的嵌套： d = { 'fruit' :{ 'apple':2 , 'bananas':6}}

访问字典值： d[ 'key' ]

字典副本： d.copy()

合并字典： d.update(d2)

合并列表： list1.extend(list2)

添加列表： list1.append(4)

字典项数： len(d)

清除字典： x.clear()

i=int(raw\_input("i= ")) 获取用户输入

if i==0:

print 'i=0'

elif i==1:

print 'i=1'

else:

print 'wrong input'

x = [1,5,2,3,4]

x.reverse() 反转排序

x.sort() 正序排序

a = [5,7,6,3,4,1,2]

b = sorted(a)sorted() 不修改原内容，可用在任何数据类型序列中，返回的总是列表形式

a

[5, 7, 6, 3, 4, 1, 2]

b

[1, 2, 3, 4, 5, 6, 7]

import time

print time.time()

1279578704.6725271

time.localtime(time.time()) 格式化时间戳为本地的时间

time.strftime('%Y-%m-%d',time.localtime(time.time()))常用

import re Python中正则表达式模块

使用re.match查找，匹配开始

match = re.match(r'匹配关键字', '文本内容')

match.group(0)

使用re.search查找，匹配任意位置

match = search(r'cat', 'dog cat dog')

match.group(0)

os.chdir(newdir) 改变工作目录

os.mkdir("test") 创建单个目录

os.rmdir("test") 删除单个目录

os.stat(file) 获取文件属性

os.system("dir") 执行操作系统命令

os.listdir(path) 列出目录下的文件

os.remove(file) 删除

os.rename(old, new) 重命名

os.curdir 返回当前目录

os.path.exists(name) 判断是否存在文件或目录name

os.path.getsize(name) 获得文件大小，若目录返回0L

os.path.splitext() 分离文件名与扩展名

os.exec(), os.execvp() 启动新进程

os.spawnv() 在后台执行程序

import os

删除文件：os.remove("文件名")

删除目录：os.removedirs（r"目录地址"）

列出文件：os.listdir()

是否文件：os.path.isfile("PATH")

重命名：os.rename（old， new）

创建多级目录：os.makedirs（r"c\python\test"）

创建单个目录：os.mkdir（"test"）

获取文件属性：os.stat（"file"）

终止当前进程：os.exit（）

分离扩展名：os.path.splitext("PATH")

获取路径名：os.path.dirname("PATH")

获取文件名：os.path.basename()

运行shell命令: os.system()

获取文件大小：os.path.getsize（filename）

读取和设置环境变量:os.getenv() os.putenv()

1.创建内存文件并写入初始数据：f = StringIO.StringIO("Hello world!")

2.读入内存文件数据：print f.read()

3.内存文件写入数据：f.write("Good day!")

4.关闭内存文件：f.close()

def calc(\*numbers): 可变参数

sum = 0

for n in numbers:

sum = sum + n \* n

return sum

def person(name, age, \*\*kw): 关键字参数

print 'name:', name, 'age:', age, 'other:', kw

def lazy\_sum(\*args): 函数做为返回值

def sum():

ax = 0

for n in args:

ax = ax + n

return ax

return sum

open(filename,参数,缓冲区大小)

r 只读方式打开文件

rb 以二进制格式打开文件用于只读

r+ 打开文件用于读写。指针在开头

rb+ 以二进制格式打开文件读写。指针在开头

w 打开文件只写入。若已存在则覆盖。若不存在则创建新文件

wb 以二进制格式打开文件仅写入。若已存在则覆盖。若不存在则创建新文件

w+ 打开文件用于读写。若已存在则覆盖。若不存在则创建新文件

a 打开文件用于追加。若该已存在则指针将会放在文件的结尾

filename.read([size]) size为读取的长度，以byte为单位

filename.readline([size]) 读一行，如果定义了size，有可能返回的只是一行的一部分

filename.readlines([size]) 把文件每一行作为一个list的一个成员，并返回这个list

filename.write(str) 把str写到文件中，write()并不会在str后加上一个换行符

filename.writelines(seq) 把seq的内容全部写到文件中(多行一次性写入)

filename.close() 关闭文件

filename.flush() 把缓冲区的内容写入硬盘

filename.ftell() 按字符数获得文件指针位置

fp.read([size]) size为读取的长度，以byte为单位

fp.readline([size]) 读一行，如果定义了size，有可能返回的只是一行的一部分

fp.readlines([size]) 把文件每行作为list的成员并返回这个list。其内部通过循环调用readline()实现。若提供size参数可能只读到文件的一部分。

fp.write(str) 把str写到文件中，write()并不会在str后加上一个换行符

fp.writelines(seq) 把seq的内容全部写到文件中(多行一次性写入)。这个函数也只是忠实地写入，不会在每行后面加上任何东西。

fp.close() 关闭文件。

fp.flush() 把缓冲区的内容写入硬盘

fp.tell() 返回文件操作标记的当前位置，以文件的开头为原点

fp.next() 返回下一行并将文件操作标记位移到下一行。把一个file用于for … in file这样的语句时，就是调用next()函数来实现

输入密码：

import getpass

PASS = getpass.getpass("请输入密码:")

print "your password is %s" %(PASS)

文件读写例子：

fp = open(file\_abs\_path,'w+')

content = 'this is a test message!!\ngood boy!\ngogo......\nhello,I\'m Hongten\nwelcome to my space!'

fp.write(content)

fp.flush()

fp.close()

print ("输出字符：%s" %("字符串"))

print ("输出整数：%d"%(25))

print ("输出浮点数：%f"%(1.83))

print ("左对齐并指定每个列的宽度:%-10s:%-8d:%-8.2f"%("Aviad",25,1.83))#右对齐使用+？

f = open('tpm.txt')

s = f.read()

print(s, '\n\n\n')

print(f.tell())

f.seek(0)

while(True):

line = f.readline()

print(line)

if(len(line) == 0):

break

f.close()

fp.close()

re.split(pattern, string, maxsplit=0)

通过正则将字串分离。若用括号将正则括起那么匹配的字串也会被列到list中返回。maxsplit是分离的次数，maxsplit=1分离一次，默认不限制

>>> re.split('\W+', 'Words, words, words.')

['Words', 'words', 'words', '']

>>> re.split('\W+', 'Words, words, words.', 1)

['Words', 'words, words.']

re.findall(pattern, string, flags=0)

找到匹配的所有子串并把它们作为列表返回。这个匹配是从左到右有序地返回。如果无匹配返回空列表。

>>> re.findall(r"\d+","12a32bc43jf3")

['12', '32', '43', '3']

字串开头是否能匹配正则表达式若不匹配返回None。

re.match("c", "abcdef")

None

数据库操作：

import MySQLdb

try:

conn=MySQLdb.connect(host='主机地址',user='用户名',passwd='密码',db='库名',port=3306)

cur=conn.cursor()

cur.execute('select \* from user')

conn.select\_db('python')

cur.execute('create table test(id int,info varchar(20))')

value=[1,'hi rollen']

try:

cur.execute('insert into test values(?,?)',value)

conn.commit()

except: 发生错误时回滚

db.rollback()

cur.close()

conn.close()

except MySQLdb.Error,e:

print "Mysql Error %d: %s" % (e.args[0], e.args[1])

查询指定对象有哪些方法：dir(int)

排序：

a = ['3','188','50','1225']

a.sort(key=int,reverse=True)

['1225', '188', '50', '3']

对象比较：

cmp(x ,y)，当x<y会返回负数、当x>y会返回正数、当x=y则返回0。

随机返回并删除一对键值：x.popitem()

>>>values = 1 , 2 , 3

>>>values

(1,2,3)

>>>a , b , c = values

>>>print a , b , c

1 2 3

删除字典：

方法一：d.pop(key)

方法二：del d[key]

调用自身的这种方法叫做递归，若达不到需要的条件则永远继续递归调用下去且程序永不停止，这种现象叫做无限递归。

例子：

def test():

test()

统计指定对象：

>>> ['a','iplaypython.com','c','b','a'].count('a')

2

语句嵌套：

if x == y:

print u'x 等于 y'

else:

if < y:

print u'x 小于 y'

else:

print u'x 大于 y'

import urllib

response = urlopen(url, data, timeout)

print response.read()

POST方式提交数据(eg)：

url = "https://passport.csdn.net/account/login?from=http://my.csdn.net/my/mycsdn"

values = {}

values['username']="1016903103@qq.com"

values['password']="XXXX"

urldata = urllib.urlencode(values)

request = urllib2.Request(url,urldata)

response = urllib2.urlopen(request)

print response.read()

GET方式提交数据(eg)：

values={}

values['username']="1016903103@qq.com"

values['password']="XXXX"

data = urllib.urlencode(values)

url = "http://passport.csdn.net/account/login"

geturl = url + "?"+data

request = urllib2.Request(geturl)

response = urllib2.urlopen(request)

print response.read()

获取结构化页面：

from bs4 import BeautifulSoup

ht=urlopen("http://www.baidu.com")

soup = BeautifulSoup(ht) 返回结构化数据

print(soup.prettify()) prettify方法添加换行和空格以让文档更清晰

soup.title输出HTML信息：<title>The Dormouse's story</title>

soup.title.name 输出HTML信息：title

soup.title.string 输出HTML信息：The Dormouse's story

soup.p 输出HTML信息：<p class="title"><b>The Dormouse's story</b></p>

soup.p["class"] 输出HTML信息：['title']

soup.find\_all('a') 输出HTML信息：[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>......

beautifulsuop模块"剖析树"的findall函数：

x = soup.findall('p') 寻找所有<P 标记，返回列表

x = soup.findall('p',id='username') 寻找id="username"的<p 标记，返回结果集

x = soup.findall('p',{'class':'img'}) 寻找<p 标记的属性或多个属性对

x = soup.findall(id=re.compile("para$")) 使用正则

x = soup.findall({'h1','h2','h3','h4'}) 返回h1~h4所有HTML标签的列表

x = soup.div.findall(....) 以导航树中div标签为起点搜索其内容(后代标签)

x = soup.div.findall(....).children 以导航树中div标签为起点搜索其内容(子标签)

for num in x:

#print(name.get\_text()) 函数get\_text()将会把文档中所有的标签都清除并返回其中的字符串

if num.name == "h2": Tag为h2类型

print num.a.string有<a..> 所以each.a.string取出内容

打印列表中所有的行：next\_silblings()用于收集表单数据

for num in soup.div.find("table",{"id":"giftlist"}).tr.next\_silblings

print(num)

读取和修改标签属性：soup.p['id'] || soup.p['key']='value'

beautifulsuop模块"剖析树"的NavigableString对象方法：

title = soup.title.next

注：soup.findAll(id="text") <==> soup.findAll("",{"id":"text"})

----------------------------------------------------------------------------------------------- image spider 1

import requests

from bs4 import BeautifulSoup

import re

DownPath = "/jiaoben/python/meizitu/pic/"

import urllib

head = {'User-Agent':'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6'}

TimeOut = 5

PhotoName = 0

c = '.jpeg'

PWD="/jiaoben/python/meizitu/pic/"

for x in range(1,4):

site = "http://www.meizitu.com/a/qingchun\_3\_%d.html" %x

Page = requests.session().get(site,headers=head,timeout=TimeOut)

Coding = (Page.encoding)

Content = Page.content#.decode(Coding).encode('utf-8')

ContentSoup = BeautifulSoup(Content)

jpg = ContentSoup.find\_all('img',{'class':'scrollLoading'})

for photo in jpg:

PhotoAdd = photo.get('data-original')

PhotoName +=1

Name = (str(PhotoName)+c)

r = requests.get(PhotoAdd,stream=True)

with open(PWD+Name, 'wb') as fd:

for chunk in r.iter\_content():

fd.write(chunk)

print ("You have down %d photos" %PhotoName)

----------------------------------------------------------------------------------------------- image spider 2

import urllib.request

import re

import time

#获取输入的帖子单页html

def getHtml2(url2):

html2=urllib.request.urlopen(url2).read().decode('utf-8')

return html2

#抽取图片相关列表，并下载图片

def gettopic(html2):

reg2=r'http://www.douban.com/group/topic/\d+'

topiclist=re.findall(reg2,html2)

x=0

#限制下载的图片数

for topicurl in topiclist:

x+=1

return topicurl

#下载图片到本地

def download(topic\_page):

reg3=r'http://img3.douban.com/view/group\_topic/large/public/.+\.jpg'

imglist=re.findall(reg3,topic\_page)

i=1

download\_img=None

for imgurl in imglist:

#取图片ID为文件名

img\_numlist=re.findall(r'p\d{7}',imgurl)

for img\_num in img\_numlist:

download\_img=urllib.request.urlretrieve(imgurl,'D:\python\code\girls\%s.jpg'%img\_num)

time.sleep(1)

i+=1

print(imgurl)

return download\_img

#调用函数

page\_end=int(input('请输入结束时的页码：'))

num\_end=page\_end\*25

num=0

page\_num=1

while num<=num\_end:

html2=getHtml2('http://www.douban.com/group/kaopulove/discussion?start=%d'%num)

topicurl=gettopic(html2)

topic\_page=getHtml2(topicurl)

download\_img=download(topic\_page)

num=page\_num\*25

page\_num+=1

else:

print('采集完成！')

-----------------------------------------------------------------------------------------------

http模拟登录：

s = requests.session()

data = {'user':'用户名','passdw':'密码'}

res=s.post('http://www.xxx.net/index.php?action=login',data);

s.get('http://www.xxx.net/archives/155/') 换成抓取的地址