



北京大学
PEKING UNIVERSITY

信息科学技术学院

实用Python程序设计

郭 炜

微信公众号



微博: <http://weibo.com/guoweiofpku>

学会程序和算法，走遍天下都不怕！



北京大学
PEKING UNIVERSITY

信息科学技术学院

玩转Python生态



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

使用Python的库



美国黄石公园

Python的库

➤ Python自带的库

`math, re, datetime, turtle, random`

➤ 无数第三方的库

`Pillow, jieba, request, matplotlib`

安装Python第三方库

- 1) 进入cmd命令行窗口
- 2) 进入安装Python的文件夹，默认通常是：

C:\Users\你的用户名\AppData\Local\Programs\Python\Python37
不知道的话查找python.exe可以找到

- 3) 再进入 scripts文件夹
- 4) `pip install 库名`

C:\WINDOWS\system32\cmd.exe

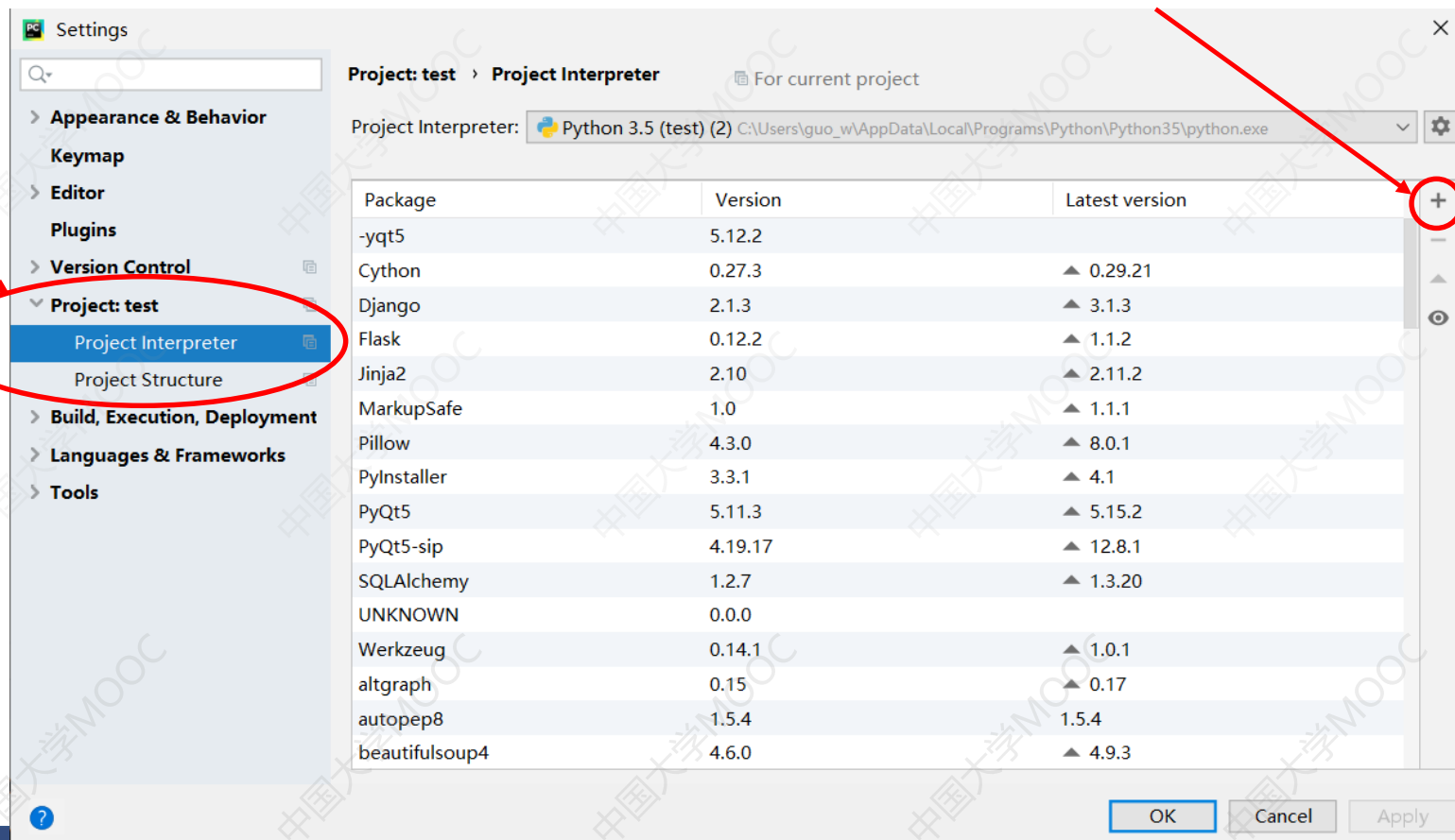
```
C:\Users\guo_w\AppData\Local\Programs\Python\Python37\Scripts>pip install pillow
```

Pycharm中安装Python第三方库

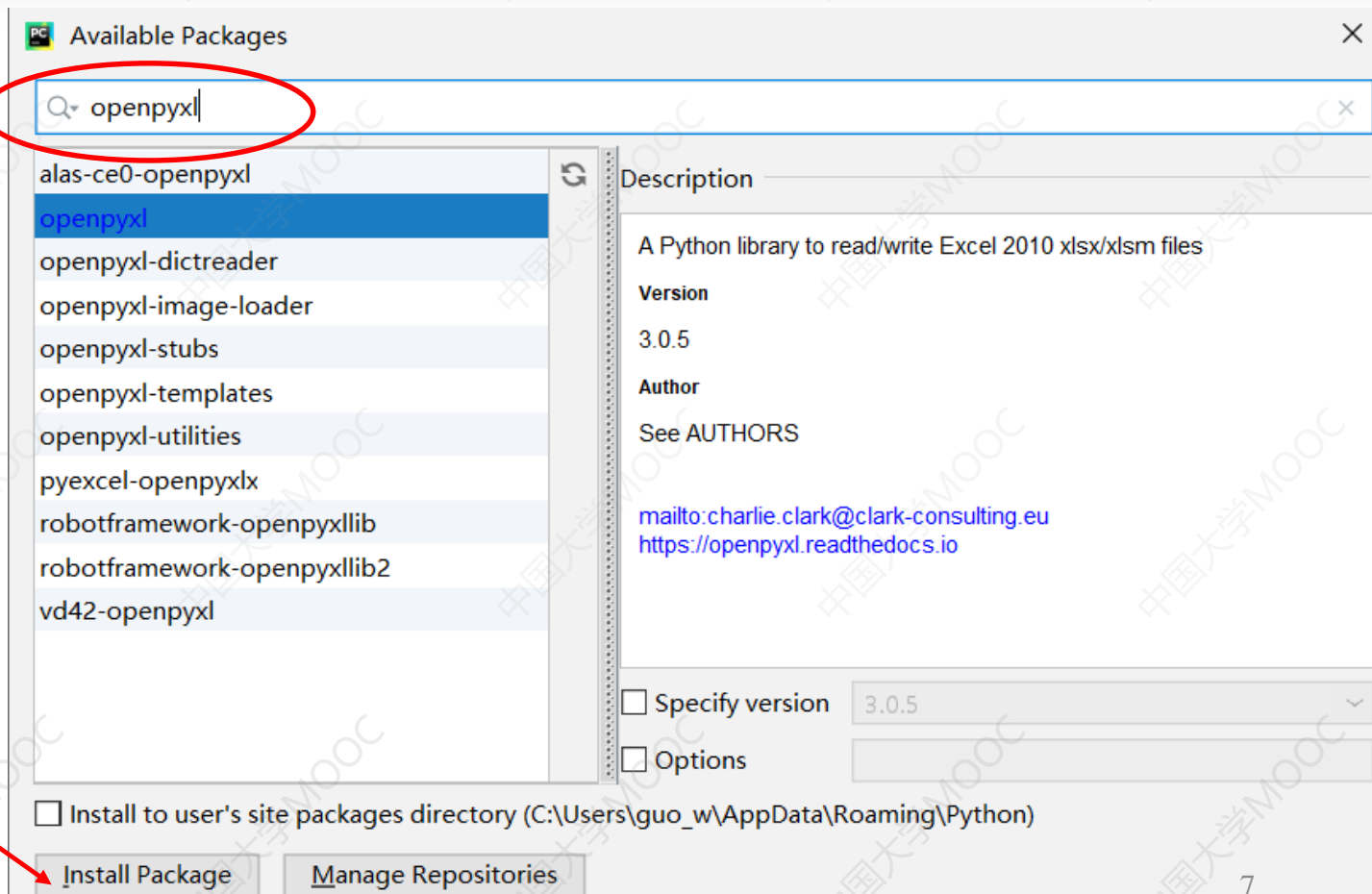
菜单: File | Settings

2. 点这里

1. 选这里



Pycharm中安装Python第三方库



1. 输入库名，回车

2. 点这里

用import导入库

```
import turtle                #turtle是一个类
turtle.setup(800,600)
turtle.fd()
```

或:

```
import turtle as tt          #此后  tt 等价于 turtle
tt.setup(800,600)
tt.fd(100)
```


import 的用法

```
import PIL.Image    #PIL.Image是类中的类  
img = PIL.Image.open("c:/tmp/pic/grass.jpg") #将图像文件载入对象img  
img.show()
```

或:

```
from PIL import Image    #从PIL库导入Image类进行图像处理  
img = Image.open("c:/tmp/pic/grass.jpg") #将图像文件载入对象img
```

import 的用法

```
import PIL.Image, PIL.ImageDraw, PIL.ImageFont  
img = PIL.Image.open("c:/tmp/pic/grass.jpg") #将图像文件载入对象img  
draw = PIL.ImageDraw.Draw(img) #以后就可以通过draw在img上画图、写字  
myFont = PIL.ImageFont.truetype("C:\\Windows\\Fonts\\simhei.ttf", 164)
```

或

```
from PIL import Image, ImageDraw, ImageFont  
img = Image.open("c:/tmp/pic/grass.jpg") #将图像文件载入对象img  
draw = ImageDraw.Draw(img) #以后就可以通过draw在img上画图、写字  
myFont = ImageFont.truetype("C:\\Windows\\Fonts\\simhei.ttf",  
164)
```

import 的用法

```
from openpyxl.styles import Font, colors, Alignment
boldRedFont = Font(size = 18, name='Times New Roman',
                    bold=True, color = colors.RED)
alignment = Alignment(horizontal='left', vertical='center')
```



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

用datetime库 处理日期、时间



阳朔遇龙河

处理日期

```
import datetime    #导入datetime模块

dtBirth = datetime.date(2000,9,27)    #创建日期对象，日期为2000年9月27日

print(dtBirth.weekday())    #>>2 输出dtBirth代表的日期是星期几。0表示星期一

dtNow = datetime.date.today()    #取今天日期，假设是 2020年8月15日

print(dtBirth < dtNow)    #>>True 日期可以比大小

life = dtNow - dtBirth    #取两个日期的时间差

print(life.days,life.total_seconds())    #>>7262 627436800.0

#两个日期相差7262天，即627436800.0秒

delta = datetime.timedelta(days = -10)    #构造时间差对象，时间差为-10天

newDate = dtNow + delta    #newDate代表的日期是dtNow的日期往前数10天

print(newDate.year,newDate.month,newDate.day,newDate.weekday())

#>>2020 8 5 2    2020年8月5日星期三

print(newDate.strftime(r'%m/%d/%Y'))    #>>08/05/2020

newDate = datetime.datetime.strptime("2020.08.05", "%Y.%m.%d")

print(newDate.strftime("%Y%m%d"))    #>>20200805
```

处理时刻

```
import datetime
```

```
tm = datetime.datetime.now() #取当前时刻，精确到微秒
```

```
print(tm.year,tm.month,tm.day,tm.hour,tm.minute,tm.second,  
      tm.microsecond)
```

```
#>>2020 8 15 20 32 53 899669 假设当前时刻是2020年8月15日20时32分53秒899669微秒
```

```
tm = datetime.datetime(2017, 8, 10, 15, 56, 10,0)
```

```
#构造一个时刻，2017年8月10日15时56分10秒0微秒
```

```
print(tm.strftime("%Y%m%d %H:%M:%S")) #>>20170810 15:56:10
```

```
print(tm.strftime("%Y%m%d %I:%M:%S %p")) #20170810 03:56:10 PM
```

```
tm2 = datetime.datetime.strptime("2013.08.10 22:31:24",  
                                  "%Y.%m.%d %H:%M:%S") #由字符串生成一个时间对象
```

```
delta = tm - tm2 #求两个时间的时间差
```

```
print(delta.days,delta.seconds,delta.total_seconds())
```

```
#>>1460 62686 126206686.0 #时间差是1460天零62686秒，总共126206686.0秒
```

处理时刻

```
delta = tm2 - tm
print(delta.days,delta.seconds,delta.total_seconds())
#>>-1461 23714 -126206686.0
delta = datetime.timedelta( days = 10, hours= 10,minutes=30,seconds=20)
#构造一个时间差, 10天10小时30分20秒
tm2 = tm + delta
print(tm2.strftime("%Y%m%d %H:%M:%S")) #>>20170821 02:26:30
```

datetime的局限

能处理的时间，年份是公元1年至9999年



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

用random库 处理随机事务



美国黄石公园

random库中的函数

<code>random.random()</code>	随机生成一个[0,1]之间的数
<code>random.uniform(x,y)</code>	随机生成一个[x,y]之间的数（含两端，下同）。x,y可以是小数
<code>random.randint(x,y)</code>	随机生成一个[x,y]之间的整数。x,y都是整数
<code>random.randrange(x,y,z)</code>	在range(x, y, z)中随机取一个数
<code>random.choice(x)</code>	从序列x中随机取一个元素。x可以是列表、元组、字符串
<code>random.shuffle(x)</code>	将列表x的元素顺序随机打乱
<code>random.sample(x,n)</code>	从序列x中随机取一个长度为n的子序列。x可以是元组、列表、集合
<code>random.seed(x)</code>	设置随机种子为x。x可以是数、元组、字符串

用法示例

```
import random

print(random.random())           #>>0.5502568034876353
print(random.uniform(1.2,7.8))  #>>5.147405813383391
print(random.randint(-20,70))    #>>20
print(random.randrange(2,20,3))  #>>17 在range(2,20,3)中随机取一个数
print(random.choice("hello,world")) #>>d
print(random.choice([1,2,'ok',34.6,'jack'])) #>>ok
lst = [1,2,3,4,5,6]
random.shuffle(lst)
print(lst)                       #>>[5, 3, 4, 2, 1, 6]
print(random.sample(lst,3))      #>>[6, 2, 3]
```

设置随机种子

```
import random
random.seed(2)    #或random.seed("ok")...种子可以是随便什么数、字符串...
#则下面程序每次运行结果都一样
print(random.random())
print(random.uniform(1.2,7.8))
print(random.randint(-20,70))
print(random.randrange(2,30,3))
print(random.choice("hello,world"))
print(random.choice([1,2,'ok',34.6,'jack']))
lst = [1,2,3,4,5,6]
random.shuffle(lst)
print(lst)
print(random.sample(lst,3))
```

实现4人玩牌的发牌模拟

```
import random

cards = [str(i) for i in range(2,11)]
cards.extend(list("JQKA"))

#cards是 ['2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K', 'A']

allCards = [] #一副牌

for s in "♠♦♥♣": #中文输入法下打“梅花”就能出梅花字符.....
    for c in cards:
        allCards.append(s+c) #allCards元素形式如: '♠3'

random.shuffle(allCards) #随机打乱52张牌

for i in range(4):
    onePlayer = allCards[i::4] #每个玩家都是隔三张牌取一张
    onePlayer.sort() #扑克牌排序规则略复杂, 这个排序不太完美
    print(onePlayer)
```

实现4人玩牌的发牌模拟

输出:

```
['♠10', '♠6', '♣5', '♣7', '♣8', '♥5', '♥7', '♥A', '♥J', '♦4', '♦6',  
'♦8', '♦K']  
['♠7', '♠8', '♠9', '♠A', '♠J', '♣9', '♣K', '♥4', '♥6', '♥K', '♦10',  
'♦5', '♦Q']  
['♠4', '♠K', '♣4', '♣Q', '♥10', '♥2', '♥3', '♥8', '♥9', '♥Q', '♦3',  
'♦9', '♦A']  
['♠2', '♠3', '♠5', '♠Q', '♣10', '♣2', '♣3', '♣6', '♣A', '♣J', '♦2',  
'♦7', '♦J']
```




北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

使用jieba进行分词



美国黄石公园

中文的分词是个问题

“买马上战场” 应该分成 “买 马 上 战场” ,
还是 “买 马上 战场” ?

不容易解决, 分词库jieba也不是总能解决。

jieba分词示例

```
import jieba                #导入分词库

s = "我们热爱中华人民共和国"

lst = jieba.lcut(s)         #分词的结果是一个列表

#默认用精确模式分词，分出的结果正好拼成原文

print(lst)                  #>>['我们', '热爱', '中华人民共和国']

print(jieba.lcut(s,cut_all = True)) #全模式分词，输出所有可能的词

#>>['我们', '热爱', '中华', '中华人民', '中华人民共和国', '华人', '人民', '人民共和国', '共和', '共和国']

print(jieba.lcut_for_search(s)) #搜索引擎模式分词

#>>['我们', '热爱', '中华', '华人', '人民', '共和', '共和国', '中华人民共和国']

s = "拼多多是个网站"

print(jieba.lcut(s))        #>>['拼', '多多', '是', '个', '网站']

jieba.add_word("拼多多")    #往词典添加新词

print(jieba.lcut(s))        #>>['拼多多', '是', '个', '网站']
```

jieba分词示例

```
s = "高克丝马微中"
```

```
print(jieba.lcut(s)) #>>['高克丝', '马微', '中']
```

```
jieba.load_userdict("c:/tmp/tmpdict.txt")
```

```
print(jieba.lcut(s)) #>>['高克', '丝马', '微中']
```

```
print(jieba.lcut("显微中，容不得一丝马虎。"))
```

```
#>>['显微', '中', '容不得', '一丝', '马虎', '。']
```

c:/tmp/tmpdict.txt文件内容如下：

高克

丝马

微中

用jieba库找出三国演义中出场次数最多的几个人

分词后对所有词进行频率统计并输出出现最多的15个词（单个字的词去掉）：

曹操 929,孔明 825,将军 756,却说 646,玄德 556,关公 508,丞相 484,二人 459,
不可 432,荆州 417,孔明曰 383,不能 380,玄德曰 380,如此 375,张飞 349,

用jieba库找出三国演义中出场次数最多的几个人

```
import jieba

f = open("c:/tmp/三国演义utf8.txt", "r", encoding="utf-8")
text = f.read()          #字符串text就是全部三国演义文本
f.close()

words = jieba.lcut(text)  #word是分出来的所有词
result = {}

for word in words:
    if len(word) == 1:
        continue
    elif word in ("诸葛亮", "孔明日"):
        word = "孔明"
    elif word in ("关公", "云长", "关云长"):
        word = "关羽"
```

用jieba库找出三国演义中出场次数最多的几个人

```
elif word in ("玄德","玄德曰"):  
    word = "刘备"  
elif word in ("孟德","操贼","曹阿瞒"):  
    word = "曹操"  
result[word] = result.get(word,0) + 1  
noneNames = {'将军','却说','荆州','二人','不可',  
'不能','如此','丞相',"商议","如何","主公","军士",  
"左右","军马","引兵","次日"} #用集合比列表、元组快  
for word in noneNames: #删除noneName中的词  
    result.pop(word)  
items = list(result.items())  
items.sort(key = lambda x : -x[1])  
for i in range(15):  
    print(items[i][0],items[i][1],end=" ,") #输出 人名 出现次数,
```

用jieba库找出三国演义中出场次数最多的几个人

输出:

孔明 1366, 刘备 1204, 曹操 973, 关羽 814, 张飞 349, 吕布 299, 孙权 264, 大喜
262, 东吴 252, 天下 252, 赵云 251, 于是 250, 今日 242, 魏兵 234, 不敢 234,



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

用openpyxl
处理excel文档



美国黄石公园

excel文档相关库

➤ office 2010前 .xls文件

用xlrd库读取

用xlwt库创建和修改

➤ office 2010及之后 .xlsx文件

用openpyxl库读写 (官网: openpyxl.readthedocs.io)

pip install openpyxl (不支持Python 3.5及以前版本)

Python 3.5及以前: pip install openpyxl==2.6.4

openpyxl读取excel文件内容

```
import openpyxl as pxl
book = pxl.load_workbook("c:/tmp/test.xlsx") #book就是整个excel文件
sheet = book.worksheets[0] #取第0张工作表
print(sheet.title) #输出工作表名字(显示于工作表下方标签)
print(sheet.min_row, sheet.max_row) #输出最小有效行号、最大有效行号
print(sheet.min_column, sheet.max_column) #输出最小有效列号、最大有效列号
for row in sheet.rows: #按行遍历整个工作表, 从第1行到sheet.max_row行(含)
    for cell in row: #遍历一行的每个单元格。cell是一个单元格
        print(cell.value) #cell.value是单元格的值, 空单元格值是None
for cell in sheet['G']: #遍历名为'G'的那一列
    print(cell.value)
for cell in sheet[3]: #遍历第3行
    print(cell.value, type(cell.value), cell.coordinate,
          cell.col_idx, cell.number_format)
```

openpyxl读取excel文件内容

```
print(px1.utils.get_column_letter(5))          #>>E 根据列号求列名
print(px1.utils.column_index_from_string('D'))  #>>4 根据列名求列号
print(px1.utils.column_index_from_string('AC')) #>>29
colRange = sheet['C:F']                        #colRange代表从第C列到第F列(含F列)
for col in colRange:                          #按列遍历第C列到第F列,col代表一列
    for cell in col:                          #cell是一个单元格
        print(cell.value)
rowRange = sheet[5:10]                        #rowRange代表第5行到第10行(含第10行)
for row in sheet['A1':'D2']:                  #按行遍历左上角是A1右下角是D2的子表
    for cell in row:                          #row[i]也可以表示第i个单元格
        print(cell.value)
print(sheet['C9'].value)                      #输出 C9单元格的值
print(sheet.cell(row=8,column=4).value)      #输出第8行第4列单元格的值
```

读取公式的计算结果

```
import openpyxl
wb = openpyxl.load_workbook("c:/tmp/style.xlsx", data_only=True)
ws = wb.worksheets[1]
print(ws['A3'].value)
```

openpyxl读取excel文件内容

➤ 获取工作表

```
sheet = book.active
```

#取活跃的工作表 (缺省就是第0张工作表)

```
sheet = book["price"]
```

#取名为" price "的工作表

➤ 遍历所有工作表，并打出其名字：

```
for sheet in book.worksheets:
```

#worksheets是工作表构成的列表

```
    print(sheet.title)
```

➤ 单元格属性

```
type(cell.value) : int, float ,str, datetime.datetime
```

```
cell.coordinate : 'A2', 'E3'
```

```
cell.col_idx : 单元格列号
```

```
cell.number_format : 数的显示格式, "General", "0.00%", "0.00E+00"等
```

openpyxl创建excel文件

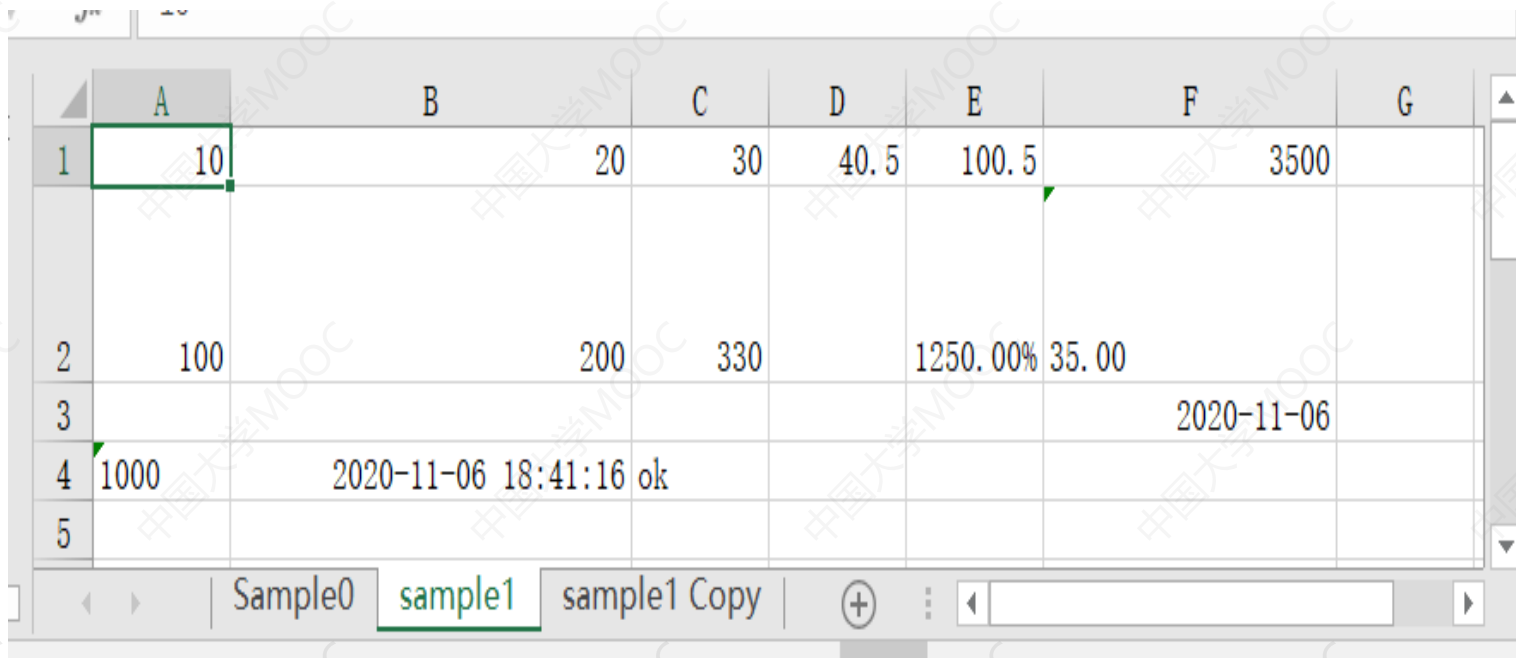
```
import openpyxl
import datetime

book = openpyxl.Workbook() #在内存创建一个excel文档, 注意w是大写
sheet = book.active #取第0个工作表
sheet.title = "sample1" #工作表取名为sample1
dataRows = ((10, 20, 30, 40.5),
             (100, 200, '=sum(A1:B2)'),
             [],
             ['1000', datetime.datetime.now(), 'ok'])
for row in dataRows:
    sheet.append(row) #在工作表中添加一行
sheet.column_dimensions['B'].width = len(str(sheet['B4'].value))
#设置B列宽度, 使其能完整显示B4单元格里的时间
sheet['E1'].value = "=sum(A1:D1)" #单元格值为公式
```

openpyxl创建excel文件

```
sheet['E2'].value = 12.5 #单元格值为小数
sheet["E2"].number_format = "0.00%" #单元格显示格式是百分比形式
sheet['F1'].value = 3500 #单元格值类型为int
sheet['F2'].value = "35.00" #单元格值类型为str
sheet['F3'].value = datetime.datetime.today().date()
sheet.column_dimensions['F'].width = len(str(sheet['F3'].value))
sheet.row_dimensions[2].height = 48 #设置第2行高度为48 points
sheet2 = book.create_sheet("Sample2") #添加名为Sample2的工作表
sheet2["A1"] = 50
sheet2 = book.create_sheet("Sample0", 0) #添加名为Sample0的工作表
sheet3 = book.copy_worksheet(sheet) #添加一张新工作表, 其为sheet的拷贝
book.remove_sheet(book["Sample2"]) #删除名为Sample2的工作表
book.save('c:/tmp/sample.xlsx') #保存文件
```

openpyxl创建excel文件



	A	B	C	D	E	F	G
1	10	20	30	40.5	100.5	3500	
2	100	200	330		1250.00%	35.00	
3						2020-11-06	
4	1000	2020-11-06 18:41:16 ok					
5							

将所有文本形式的数转换为真正的数

```
import openpyxl as pxl
book = pxl.load_workbook("c:/tmp/test2.xlsx")
for sheet in book.worksheets:
    for row in sheet.rows:
        for cell in row:
            v = cell.value
            if type(v) == str:
                if v.isdigit():                #如果v全部由数字组成
                    cell.value = int(v)
                else:
                    try:
                        cell.value = float(v) #如果不是小数格式, 转换会引发异常
                    except: pass
book.save("c:/tmp/test3.xlsx")
```


将真正的数转换为文本形式

```
import openpyxl as pxl
book = pxl.load_workbook("c:/tmp/test2.xlsx")
for sheet in book.worksheets:
    for row in sheet.rows:
        for cell in row:
            if type(cell.value) == int or type(cell.value) == float:
                cell.value = str(cell.value)
book.save("c:/tmp/test3.xlsx")
```

openpyxl指定单元格的样式

```
import openpyxl
from openpyxl.styles import
Font, colors, PatternFill, Alignment, Side, Border
book = openpyxl.Workbook() #在内存创建一个excel文档, 注意w是大写
sheet = book.active #取第0个工作表
for i in range(4): #添加4行5列数据
    sheet.append([i*5 + j for j in range(5)])
side = Side(style="thin") #边线类型, 还可以是 "thick", "medium", "dotted"等
border = Border(left=side, right=side, top=side, bottom=side) #边框类型
for row in sheet.rows:
    for cell in row:
        cell.border = border #为单元格设置边框类型
sheet['A1'].fill = PatternFill(patternType='solid', fgColor="00ff00")
#单元格底色设置为绿色, 00ff00表示红色分量0, 绿色分量255, 蓝色分量0
```

openpyxl指定单元格的样式

```
a1 = sheet['A1']  
italicRedFont = Font(size = 18, name='Times New Roman', bold=True,  
color = colors.RED)  
a1.font = italicRedFont #设置单元格字体  
sheet['A2'].font = sheet['A1'].font.copy(italic = True)  
#A2的字体和A1的字体一样, 但是是斜体  
sheet.merge_cells('C2:D3') #从C2到D3合并为一个单元格, 此后名为 c2  
sheet['C2'].alignment = Alignment(horizontal='left', vertical='center')  
#c2文字水平左对齐, 垂直居中  
book.save("c:/tmp/style.xlsx")
```

openpyxl指定单元格的样式

✓ f_x		0					
✕		A	B	C	D	E	F
	1	0	1	2	3	4	
	2	5	6	7		9	
	3	10	11			14	
	4	15	16	17	18	19	
	5						
	6						

xlrd读取excel文件内容

c:\tmp\sample.xlsx

	A	B	C	D	E
1	姓名	资产 (亿)			
2	马云	2000			
3	马化腾	2100			
4					
5					
6					
7					
8					

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G
1	姓名	学号					
2	王二	12383					
3	赵三	48884					
4							
5							
6							
7							
8							

The formula bar shows the value 48884. The status bar at the bottom shows '就绪' (Ready) and '100%' zoom.

```
import xlrd

book = xlrd.open_workbook("c:\\tmp\\sample.xlsx" ) #打开指定文件
for s in book.sheets(): #遍历所有的工作表
    print (s.name) #打出工作表名字
sheet1 = book.sheet_by_index(0) #通过sheet索引获得sheet对象
sheet1_name = book.sheet_names()[0] # 获得指定索引的sheet名称
print(sheet1_name) #>>富豪记录
sheet1 = book.sheet_by_name(sheet1_name) #通过名字获得sheet对象
nrows = sheet1.nrows #总行数
ncols = sheet1.ncols #总列数
for i in range(nrows): #打印表中的内容
    for j in range(ncols):
        cell_value = sheet1.cell_value(i, j)
        print(cell_value, end = "\t")
    print("")
```

```
import xlrd

book = xlrd.open_workbook("c:\\tmp\\sample.xlsx" ) #打开指定文件
for s in book.sheets(): #遍历所有的工作表
    print (s.name) #打出工作表名字
sheet1 = book.sheet_by_index(0) #通过sheet索引获得sheet对象
sheet1_name = book.sheet_names()[0] # 获得指定索引的sheet名称
print(sheet1_name) #>>富豪记录
sheet1 = book.sheet_by_name(sheet1_name) #通过名字获得sheet对象
nrows = sheet1.nrows #总行数
ncols = sheet1.ncols #总列数
for i in range(nrows): #打印表中的内容
    for j in range(ncols):
        cell_value = sheet1.cell_value(i, j)
        print(cell_value, end = "\t")
    print("")
```

输出:

富豪记录

学生记录

富豪记录

姓名	资产 (亿)
----	--------

马云	2000.0
----	--------

马化腾	2100.0
-----	--------

xlwt创建excel文件

创建：c:\\tmp\\sample2.xls

样式						
A1					姓名	
A	B	C	D	E	F	G
1	姓名	绩点				
2	张三ffff	3.4				
3	赵二	3.9				
4						
5						
6						
7						
8						
成绩单		名单				
就绪						

样式

A1

✕

✓

f_x

学号

编辑栏

A	B	C	D	E	F	G
1	学号	姓名				
2	1234	Jack				
3	6656	Jone				
4						
5						
6						
7						
8						

成绩单

名单

+

就绪

-

+

100%

```
import xlwt
```

#创建一个Workbook对象, 即创建一个Excel文件

```
book = xlwt.Workbook(encoding = "utf-8", style_compression = 0)
```

#创建一个工作表

```
sheet = book.add_sheet("成绩单", cell_overwrite_ok = True)
```

#向表sheet中添加数据

```
sheet.write(0, 0, "姓名") #在第0行0列的单元格写入 "姓名"
```

```
sheet.write(0, 1, "绩点")
```

```
sheet.write(1, 0, "王二")
```

```
sheet.write(1, 1, "3.4")
```

```
sheet.write(2, 0, "赵二")
```

```
sheet.write(2, 1, "3.9")
```

```
sheet = book.add_sheet("名单", cell_overwrite_ok = True)
sheet.write(0, 0, "学号")
sheet.write(0, 1, "姓名")
sheet.write(1, 0, "1234")
sheet.write(1, 1, "Jack")
sheet.write(2, 0, "6656")
sheet.write(2, 1, "Jone")
book.save("c:\\tmp\\sample2.xls") #sample2.xls如果本来存在, 会被覆盖
```

xlwt向单元格添加公式

```
import xlwt
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('My Sheet')
worksheet.write(0, 0, 5) # Outputs 5
worksheet.write(0, 1, 2) # Outputs 2
worksheet.write(1, 0, xlwt.Formula('A1*B1'))
# Should output "10" (A1[5] * A2[2])
worksheet.write(1, 1, xlwt.Formula('SUM(A1,B1)'))
# Should output "7" (A1[5] + A2[2])
workbook.save('c:\\tmp\\Excel_Workbook.xls')
```

xlwt向单元格添加日期

```
import xlwt
import datetime
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('My Sheet')
style = xlwt.XFStyle()
style.num_format_str = 'M/D/YY'
# Other options:
#D-MMM-YY, D-MMM, MMM-YY, h:mm, h:mm:ss,
#h:mm, h:mm:ss, M/D/YY h:mm, mm:ss, [h]:mm:ss, mm:ss.0
worksheet.write(0, 0, datetime.datetime.now(), style)
workbook.save('Excel_Workbook.xls')
```

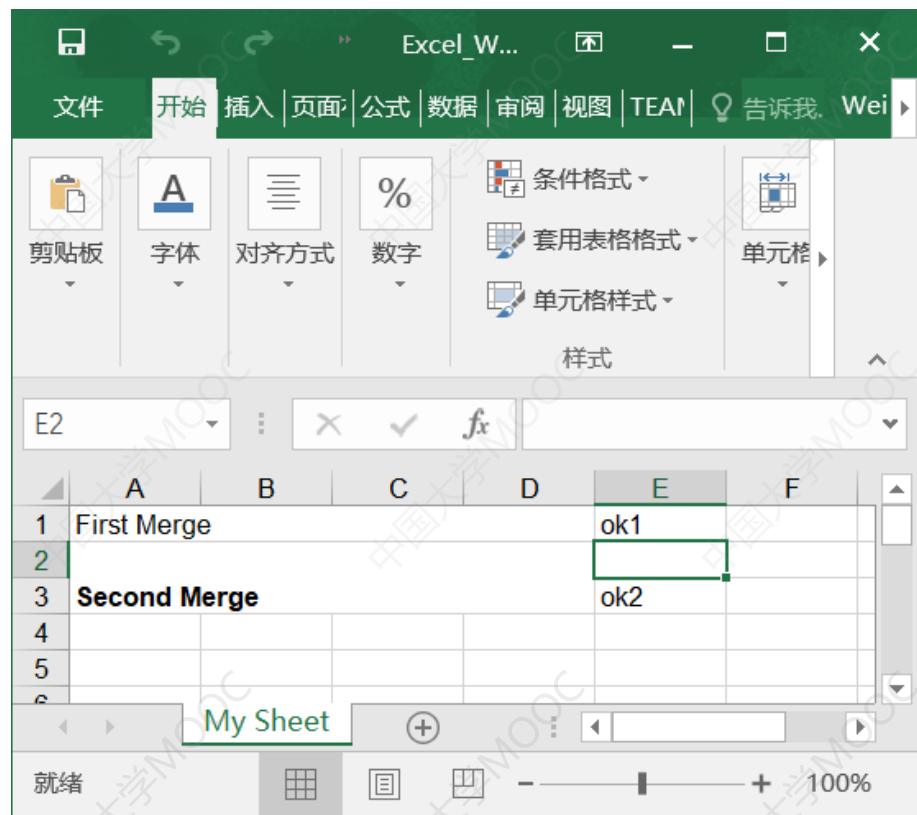
xlwt向单元格添加一个超链接

```
import xlwt
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('My Sheet')
worksheet.write(0, 0,
               xlwt.Formula('HYPERLINK("http://www.pku.edu.cn"; "PKU") '))
# Outputs the text "Google" linking to http://www.google.com
workbook.save('Excel_Workbook.xls')
```

xlwt合并单元格

```
import xlwt
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('My Sheet')
worksheet.write_merge(0, 0, 0, 3, 'First Merge')
worksheet.write(0, 4, "ok1")
# Merges row 0's columns 0 through 3.
font = xlwt.Font() # Create Font
font.bold = True # Set font to Bold
style = xlwt.XFStyle() # Create Style
style.font = font # Add Bold Font to Style
worksheet.write_merge(1, 2, 0, 3, 'Second Merge', style)
# Merges row 1 through 2's columns 0 through 3.
worksheet.write(2, 4, "ok2")
workbook.save('c:\\tmp\\Excel_Workbook.xls')
```

xlwt合并单元格





北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

用Pillow处理图像



美国黄石公园

用Pillow处理图像

```
pip install pillow
```

➤ 图像缩放和旋转

➤ 图像加滤镜

➤ 图像切割

➤ 图像加水印

➤ 图像素描化

➤ 图像加文字

图像的常识

➤ 图像由像素构成

屏幕上每个像素由3个距离非常近的点构成，分别显示红、绿、蓝三种颜色，每个像素可以由一个元组 (r, g, b) 表示， r, g, b 通常是不超过255的整数

➤ 图像模式：

RGB：一个像素有红、绿、蓝三个分量

RGBA：一个像素有红、绿、蓝三个分量，以及透明度分量

CYMK：一个像素有青色（Cyan）、洋红色（Magenta），黄色（Yellow），黑色（K代表黑）四个分量，即每个像素用元组 (c, y, m, k) 表示，对应于彩色打印机或者印刷机的4种颜色的墨水。

L：黑白图像。每个像素就是一个整数，代表灰度。

图像的缩放

#注意, 安装时 `pip install pillow`

`from PIL import Image` **#导入Image类进行图像处理**

`img = Image.open("c:/tmp/pic/grass.jpg")` **#将图像文件载入对象img**

`w,h = img.size` **#获取图像的宽和高 (单位: 像素), img.size是个元组**

`newSize = (w//2,h//2)` **#生成一个新的图像尺寸**

`newImg = img.resize(newSize)` **#得到一张原图像一半大小的新图像**

`newImg.save("c:/tmp/pic/grass_half.jpg")` **#保存新图像文件**

`newImg.thumbnail((128,128))` **#变成宽高各128像素的缩略图**

`newImg.save("c:/tmp/pic/grass_thumb.png", "PNG")`

#保存新图像文件为png文件

`newImg.show()`

#显示图像文件

图像的旋转、翻转图像、和滤镜效果

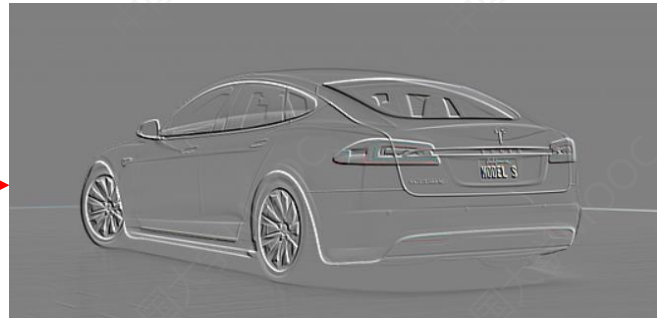
```
from PIL import Image
from PIL import ImageFilter
img = Image.open("c:/tmp/pic/grass_half.jpg")
print(img.format, img.mode)
newImg = img.rotate(90, expand = True)
newImg.show()
newImg = img.transpose(Image.FLIP_LEFT_RIGHT)
newImg = img.transpose(Image.FLIP_TOP_BOTTOM)
newImg = img.filter(ImageFilter.BLUR)
```

#实现滤镜效果需要
#>>JPEG RGB
#图像逆时针旋转90度
#左右翻转
#上下翻转(颠倒)
#模糊效果

滤镜效果:

ImageFilter.CONTOUR
ImageFilter.EDGE_ENHANCE
ImageFilter.EMBOSS
ImageFilter.SMOOTH
ImageFilter.SHARPEN

轮廓效果
边缘增强
浮雕
平滑
锐化



图像的裁剪

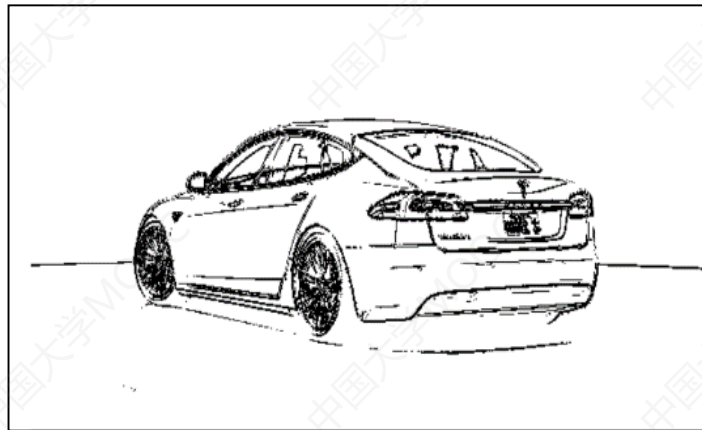


图像的裁剪

```
from PIL import Image
img = Image.open("c:/tmp/pic/grass.jpg")
w,h = img.size[0]//3,img.size[1]//3
gap = 10                                #九宫图中相邻两幅子图间的空白宽10像素
newImg = Image.new("RGB", (w * 3 + gap * 2,
                           h * 3 + gap * 2), "white")

for i in range(0,3):
    for j in range(0,3):
        clipImg = img.crop((j*w,i*h, (j+1)*w, (i+1)*h))
        clipImg.save("c:/tmp/pic/grass%d%d.jpg" % (i,j))
        newImg.paste(clipImg, (j*(w + gap), i * (h + gap)))
newImg.save("c:/tmp/pic/grass9.jpg")    #保存九宫图
newImg.show()
```

图像的素描化



图像的素描化

```
from PIL import Image
def makeSketch(img, threshold):
    w, h = img.size
    img = img.convert('L') #图像转换成灰度模式
    pix = img.load() #获取像素矩阵
    for x in range(w-1):
        for y in range(h-1):
            if abs(pix[x,y] - pix[x+1,y+1]) >= threshold:
                pix[x,y] = 0
            else:
                pix[x,y] = 255
    return img
img = Image.open("c:/tmp/pic/models2.jpg")
img = makeSketch(img, 15) #阈值threshold为15
img.show()
```

给图像添加水印



北京大学
PEKING UNIVERSITY



给图像添加水印

- 原理: paste时可以用“掩膜”指定img的每个像素粘贴过去的透明度。如果透明度为0, 则完全透明, 如果透明度为255,则完全遮盖imgSrc原来的像素。
- mask参数即为掩膜, 是个模式为"L"的图片(Image对象)

`imgSrc.paste (img, (x,y) , mask = msk)`



给图像添加水印

```
from PIL import Image
def getMask(img,isTransparent,alpha):
    #返回由img变出来的掩膜
    if img.mode != "RGBA":
        img = img.convert('RGBA')    #转换成RGBA模式的图像
    w, h = img.size
    pixels = img.load()               #获取像素矩阵
    for x in range(w):
        for y in range(h):
            p = pixels[x,y]           #p是一个四元素元组(r,g,b,a)
            if isTransparent(p[0],p[1],p[2]): #判断p是否应该变成透明像素
                #p[0],p[1],p[2] 分别是红、绿、蓝分量
                pixels[x,y] = (p[0],p[1],p[2],0)
            else:
                pixels[x,y] = (p[0],p[1],p[2],alpha)
    r, g, b, a = img.split()         # 分离出img中的四个分量,a就是掩膜
    return a
```

给图像添加水印

```
img = Image.open("c:/tmp/pic/pku.png")
msk = getMask(img,
               lambda r,g,b: r > 245 and g > 245 and b > 245, 130)
imgSrc = Image.open("c:/tmp/pic/iceland1.png ")
imgSrc.paste(img, (imgSrc.size[0] - img.size[0] - 30 ,
                  imgSrc.size[1] - img.size[1] - 30), mask = msk)
#粘贴透明图像img到imgSrc的右下角，用a做掩膜
imgSrc.show()
```