



普通高等院校“十三五”规划教材



上海市高等职业院校课程规划教材（二）参考教材

Python程序设计基础 (第2版)

主编 李东方 文欣芳

中国工信出版集团 电子工业出版社
http://www.pitp.com.cn

第12章 基于第三方库的 应用举例

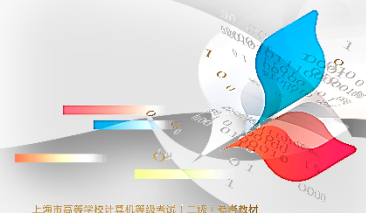
本章教学目标:

- 了解利用openpyxl等第三方库对Excel的操作
- 了解利用Pillow等第三方库对图片的操作

上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

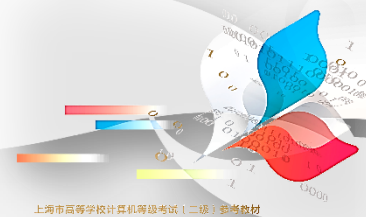
(第2版)

主 编 李东方 支晓勇

12.1 对Excel的操作

- 常见的Python第三方库

- xlrd, xlwt, xlutils, pyExcelerator, openpyxl, xlswriter
 - xlrd, xlwt, xlutils和pyExcelerator可处理.xls文件, 网上资源较多。
 - xlrd, xlwt分别司职读和写
 - xlutils和pyExcelerator读/写均可, 但功能偏少。
 - openpyxl和xlswriter可处理.xlsx文件
 - Openpyxl读/写均可, 能满足基本操作要求
 - Xlswriter写操作功能非常专业, 但没有读取功能



12.1 对Excel的操作

```

C:\WINDOWS\system32>pip install openpyxl
Collecting openpyxl
  Downloading openpyxl-3.0.3.tar.gz (172 kB)
    | 81 kB 2.3 kB/s eta 0:00:3
    | 92 kB 2.4 kB/s eta 0:00:
    | 102 kB 2.5 kB/s eta 0
    | 112 kB 2.7 kB/s eta
    | 122 kB 2.7 kB/s e
    | 133 kB 2.8 kB/s
    | 143 kB 3.1 kB
    | 153 kB 2.8
    | 163 kB 2.
    | 172 kB 2
.3 kB/s
Collecting jdcal
  Downloading jdcal-1.4.1-py2.py3-none-any.whl (9.5 kB)
Collecting et_xmlfile
  Downloading et_xmlfile-1.0.1.tar.gz (8.4 kB)
Installing collected packages: jdcal, et-xmlfile, openpyxl
  Running setup.py install for et-xmlfile ... done
  Running setup.py install for openpyxl ... done
Successfully installed et-xmlfile-1.0.1 jdcal-1.4.1 openpyxl-3.0.3

C:\WINDOWS\system32>
    
```

12.1.1 用openpyxl在内存中创建工作表

- 在内存中创建工作簿，或将已有工作簿装入内存：

```
from openpyxl import Workbook
```

```
wb = Workbook() #创建
```

```
wb=openpyxl.load_workbook(filename=<xlsx文件路径及名称>) #打开
```

由于Excel文件具有独占性，因此只能访问不会同时被其他程序打开的.xlsx文件。

- 新创建的空工作簿应创建新的工作表

```
ws1 = wb.create_sheet("Mysheet")
```

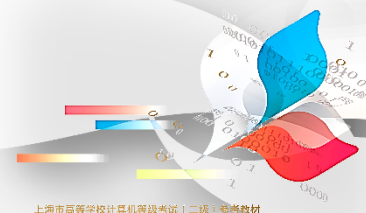
```
ws1 = wb.create_sheet("Mysheet", 0) # 将其放在所有工作表的首位
```

- 对于已有工作簿可选择要操作的工作表对象

```
openpyxl.sheet_ranges=wb['Sheet1'] # 注意工作表名称的大小写
```

- 用openpyxl.workbook.Workbook.active()函数可将最后创建的工作表或指定的工作表作为活动工作表

```
ws = wb.active
```



12.1.2 数据的读取

- 使用openpyxl，可以直接访问活动工作表的单元格

```
cell_A4=ws['A4']
```

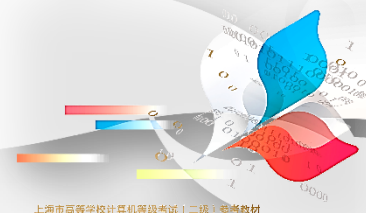
cell_A4对象实例的类型是单元格，其值为cell_A4.value。

```
<class 'openpyxl.cell.cell.Cell'>
```

- 也可以直接以行、列定位访问单元格，读取数据或赋值

```
cell_B4= ws.cell(row=4, column=2, value='上海市')
```

注意：这里的行、列参数row和 column都是从1起始的，与Python其他对象的指针下标不一致。



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主編 李东方 支晓勇

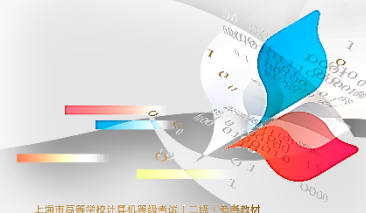
12.1.2 数据的读取

- 遍历整个工作表的所有单元格

使用`openpyxl.worksheet.Worksheet.rows()`方法,并同时用`tuple()`函数转换为元组类型,即可得到以每行元组为元素的二维元组。

```
>>> tuple(ws.rows)
```

```
((<Cell Sheet.A1>, <Cell Sheet.B1>, <Cell Sheet.C1>),  
(<Cell Sheet.A2>, <Cell Sheet.B2>, <Cell Sheet.C2>),  
(<Cell Sheet.A3>, <Cell Sheet.B3>, <Cell Sheet.C3>),  
(<Cell Sheet.A4>, <Cell Sheet.B4>, <Cell Sheet.C4>),  
(<Cell Sheet.A5>, <Cell Sheet.B5>, <Cell Sheet.C5>),  
(<Cell Sheet.A6>, <Cell Sheet.B6>, <Cell Sheet.C6>),  
(<Cell Sheet.A7>, <Cell Sheet.B7>, <Cell Sheet.C7>),  
(<Cell Sheet.A8>, <Cell Sheet.B8>, <Cell Sheet.C8>),  
(<Cell Sheet.A9>, <Cell Sheet.B9>, <Cell Sheet.C9>))
```



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主編 李東方 支原勇

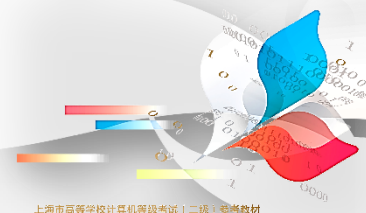
12.1.2 数据的读取

- 遍历整个工作表的所有单元格

使用`openpyxl.worksheet.Worksheet.columns()`方法, 将其用`tuple()`函数转换为元组类型, 即可得到以每列元组为元素的二维元组。

```
>>> tuple(ws. columns)
```

```
((<Cell Sheet.A1>,<Cell Sheet.A2>,<Cell Sheet.A3>,<Cell Sheet.A4>,<Cell Sheet.A5>,<Cell Sheet.A6>,<Cell Sheet.A7>,<Cell Sheet.A8>,<Cell Sheet.A9>), ..., (<Cell Sheet.C1>,<Cell Sheet.C2>,<Cell Sheet.C3>,<Cell Sheet.C4>,<Cell Sheet.C5>,<Cell Sheet.C6>,<Cell Sheet.C7>,<Cell Sheet.C8>,<Cell Sheet.C9>))
```

上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主编 李东方 支晓勇

12.1.2 数据的读取

- 读取工作表中的部分数据
 - 使用单元格区间
`cell_range = ws['A1:C2']`
 - 使用列区间
`col_range = ws['C:D']`
 - 使用行区间
`row_range = ws[5:10]`
 - 有效行数
`len(tuple(ws.rows))`
 - 有效列数
`len(tuple(ws.columns))`

12.1.3 编辑操作和写入

- 赋值

Excel的单元格除可以赋值为字符串、整数和浮点数之外，还可以接收日期、百分数、公式等赋值

```
ws['A1'] = datetime.datetime(2016, 9, 18)
ws['B1'] = '0.15%' # 需打开后手工转为数字
ws['C1'] = '0000001234' # 自动判别为字符格式
ws['D1'] = '=SUM(D2:D10)' # 表达式的写法与Excel
                           语法一致
```

若数据以每行元组（或列表）为元素的二维元组（或列表）形式存在，则可利用循环语句将数据写入活动工作表中。

- 【例12-1】



12.1.3 编辑操作和写入

- 赋值

Excel的单元格除可以赋值为字符串、整数和浮点数之外，还可以接收日期、百分数、公式等赋值

```
ws['A1'] = datetime.datetime(2016, 9, 18)
```

```
ws['B1'] = '0.15%' # 需打开后手工转为数字
```

```
ws['C1'] = '0000001234' # 自动判别为字符格式
```

```
ws['D1'] = '=SUM(D2:D10)' # 表达式的写法与Excel  
语法一致
```

若数据以每行元组（或列表）为元素的二维元组（或列表）形式存在，则可利用循环语句将数据写入活动工作表中。





12.1.3 编辑操作和写入

• 合并与拆分单元格

工作表对象的`merge_cells()`和`unmerge_cells()`方法可使用单元格名称字符串参数或起止行列作为参数

```
ws.merge_cells('A1:B1')
```

```
ws.unmerge_cells('A1:B1')
```

```
ws.merge_cells(start_row=2,start_column=1,end_row=2,end_column=4)
```

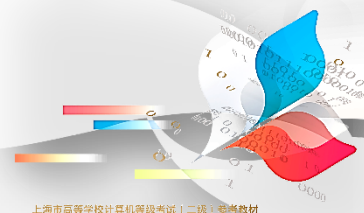
```
ws.unmerge_cells(start_row=2,start_column=1,end_row=2,end_column=4)
```

• 单元格格式化

引用`openpyxl.styles`, 可利用`Font` (字体), `Color` (颜色)、`Border` (加粗), `Side` (边缘), `PatternFill` (填充), `Alignment` (排列), `Protection` (锁定) 等函数设定单元格的格式化属性。

```
ws['A1'].font.italic = True # 为A1单元格字符格式设为斜体
```

```
ws['B1'].font = Font(name='黑体', size=14, color=colors.RED, italic=True)
```



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主 编 李东方 支晓勇

12.1.3 编辑操作和写入

• 图表

引用`openpyxl.chart`可创建2D或3D图表。以直方图为例，用`chart = BarChart()`实例化直方图对象，用`Reference()`函数实例化图表数据对象和坐标轴数据对象，用`chart.add_data()`方法数据绘图，其范围选择符合Excel的语法习惯，用`chart.set_categories()`方法添加坐标数据，最后用`add_chart()`方法将图表插入活动工作表对象的指定起始单元格。

• 保存

用`wb.save (<.xlsx文件路径及名称>)`即可覆盖性写入文件。同样，由于Excel文件的独占性，只能写入未同时被其他程序打开的.xlsx文件。

• 【例12-1】、【例12-2】



12.2 图片操作

- PIL (Python Imaging Library) 是著名的图片处理第三方库，包含基本的图像处理、特效合成、滤镜等方法。其替代库Pillow修复了PIL的一些缺陷，提供了对Python3的支持。
- 在连接互联网的状态下，用
`"C:\Python3\Scripts\easy_install.exe pillow"` 命令即可快速安装PIL (Pillow) 库。Pillow库中除核心模块Image外，还包含ImageChops (图片计算)、ImageEnhance (图片效果)、ImageFilter (滤镜)、ImageDraw (绘图) 等主要模块。



12.2.1 Image模块及应用

- 图片打开函数

`open(file, openmode)`

参数file和openmode分别为文件名和打开方式（默认为'r'，只读）。

- 新建图片函数

`new(mode, size, color=0)`

mode为图片色彩模式，取值为：'RGB'、'CMYK'、'LAB'等；
size为图片大小，是由水平像素数与垂直像素数组成的元组；
color颜色，默认为黑色（0）。

- 复制图片函数

`copy()`

将图片对象复制到内存中。

- 粘贴图片函数

`paste(region, size)`

将内存中的图片粘贴到对象region中。其中，region为粘贴对象，size为图片大小。

12.2.1 Image模块及应用

- 显示图片

`show()`

显示内存中图片。

- 重新设置大小

`resize(size)`

`size`为图片大小，是水平像素与垂直像素元组。

- 图片旋转

`rotate(angle)`

`angle`为旋转角度。

- 图片变换

`transpose(method)`

`method`可选项有：`FLIP_LEFT_RIGHT`（左右镜像）、`FLIP_TOP_BOTTOM`（上下镜像）、`ROTATE_90`（顺时针90°）、`ROTATE_180`（顺时针180°）、`ROTATE_270`（逆时针90°）等。



(9) 图片裁剪

`crop((x1,y1,x2,y2))`

其中，各参数为自左上角点至右下角点的坐标元组。

(10) 缩略图

`thumbnail(size)`

其中，size为缩略图大小。

(11) 转换函数

`convert(mode)`

功能是转换色彩模式等。其中，mode可取值为：L、RGB、CMYK等。

(12) 混合图片函数

`blend(im1,im2,alpha)`

其中，im1、im2分别相当于Photoshop中参加混合的上、下两层图片；alpha为混合透明度，通过 $im1 * (1 - alpha) + im2 * alpha$ 得到混合结果。

(13) 遮罩图片函数

`composite(im1,im2,mask)`

其中，im1相当于Photoshop中的绑定遮罩层的图片；

im2相当于下层图片（被遮罩）；

mask为遮罩层图片，要求为黑白二值、灰度或RGBA色彩模式。

(14) 保存图片

`save(file,format)`

功能是将内存中的图片写入文件中。

12.2.2 ImageChops特效与合成

- ImageChops模块包含一些通过计算图片通道中的像素值而进行特效合成的函数，相当于Photoshop中的图层特效



(1) 正片叠底

`multiply(im1,im2)`

这类似于在同一光源叠放两张胶片的投影结果，高亮度视为透明

```
from PIL import Image
from PIL import ImageChops
im1=Image.open('c:/1.jpg')
im2=Image.open('c:/2.jpg')
m=ImageChops.multiply(im1,im2)
m.show()
m.save('c:/multiply.jpg')
```





(2) 滤色

`screen(im1,im2)`

这类似于两张胶片分别透过不同光源在同一屏幕上的投影结果，低亮度视为透明

```
from PIL import Image
from PIL import ImageChops
im1=Image.open('c:/1.jpg')
im2=Image.open('c:/2.jpg')
m=ImageChops.multiply(im1,im2)
m.show()
m.save('c:/screen.jpg')
```



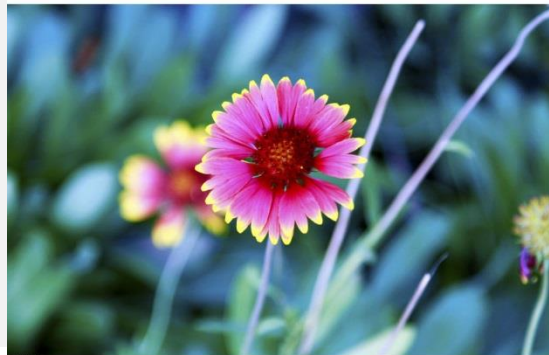


(3) 反相

`invert(im1)`

类似于照相底片的效果，以255减去像素的色彩值后得到新的色彩值

```
from PIL import Image
from PIL import ImageChops
im1=Image.open('c:/1.jpg')
m=ImageChops.invert(im1)
m.show()
m.save('c:/inv1.jpg')
```





(4) 相减

```
subtract(im1,im2)
```

其求得两张图片对应像素值之差，用于采集相同背景图片上的差异前景。

```
from PIL import Image
from PIL import ImageChops
im1=Image.open('c:/1.jpg')
im2=Image.open('c:/2.jpg')
m=ImageChops.subtract(im1,im2)
m.show()
m.save('c:/subtr.jpg')
```

