

# python 模块介绍-xlwt 创建 xls 文件（excel）

## 目录

项目简介.....	1
<b>Xlwt 简介:</b> .....	2
快速入门.....	3
插入图片.....	5
设置样式.....	6
公式.....	7
合并单元格.....	8
日期格式.....	8
边框.....	9
字体颜色.....	10
超级链接.....	11
编码.....	12
冻结.....	12
数值格式.....	14
更多公式.....	15
保护.....	16
综合实例.....	19

## 项目简介

Python 中文库 <https://bitbucket.org/xurongzhong/python-chinese-library> 主要基于个人的使用经验，收集一些重要的外部和内部模块的中文教程和实例。发起人是 [ouyangchongwu@gmail.com](mailto:ouyangchongwu@gmail.com)，[xurongzhong@gmail.com](mailto:xurongzhong@gmail.com)。

欢迎大家加入分享经验。联系方式: [xurongzhong@gmail.com](mailto:xurongzhong@gmail.com), 微博: <http://weibo.com/cizhenshi>, python 及测试开发 qq 群 1: 113938272, 群 2:6089740。

文件下载:

- 1, <https://bitbucket.org/xurongzhong/python-chinese-library/downloads> 下载。 推荐
- 2, hg clone 克隆所有文件 hg clone <https://bitbucket.org/xurongzhong/python-chinese-library>。
- 3, <https://bitbucket.org/xurongzhong/python-chinese-library/src> 浏览文件, 右键点击文件, 选另存为下载。

Bug 提交: <https://bitbucket.org/xurongzhong/python-chinese-library/issuest>。

版本管理

版本号	修订发布时间	修订人	备注
V1.0	2013-11-15	<a href="mailto:ouyangchongwu@gmail.com">ouyangchongwu@gmail.com</a>	初始版本, 由 xlwt 的实例参考生成。
V1.0.1	2013-11-18	<a href="mailto:xurongzhong@gmail.com">xurongzhong@gmail.com</a>	添加 csv 模块, 修改地址为 bitbucket。

参考资料:

下载地址: <https://pypi.python.org/pypi/xlwt/0.7.5>

官方网址: <http://www.python-excel.org/>

介绍胶片: <http://www.simplistix.co.uk/presentations/python-excel.pdf>

### **Xlwt 简介:**

功能: 用于生成 97/2000/XP/2003 xls 文件。

月下载量: 3 万左右

Python 版本: *Python 2.3 to 2.7*

当前版本: 0.7.5

下载地址: <http://pypi.python.org/pypi/xlwt>

平台：跨平台

相关模块：

csv python 标准模块。推荐。

xlrd 读取 .xls, .xlsx 文件，月下载 3.5 万左右，推荐。

xlutils xlrd 和 xlwt 的集合，月下载 0.7 万左右，推荐。

openpyxl 读写 Excel 2007 xlsx/xlsm 文件 月下载 1.5 万左右，推荐。纯 python，效率不高。

Matplotlib 2D 作图模块，适用于基于 excel 作图，推荐。月下载 2 千左右。

Pywin32 python windows 扩展 月下载 2.5 万左右，不跨平台，通过 COM 口连接 excel。

Pyxll 在 excel 中使用 python 替代 vbs。<http://www.pyxll.com/>。推荐。类似模块有 pyinex。

Python For Excel <http://www.opentradingsystem.com/PythonForExcel/main.html>

XlsxWriter 写 xlsx 文件，月下载 6k 左右，推荐。纯 python。

## 快速入门

下面例子，创建一个名为 mini.xls 的文件，它有一个空 sheet: 'xlwt was here'。代码见 mini.py。

```
from xlwt import *

w = Workbook()
ws = w.add_sheet('xlwt was here')
w.save('mini.xls')
```

Workbook 类初始化时有 encoding 和 style\_compression 参数。

encoding，设置字符编码，一般要这样设置：w = Workbook(encoding='utf-8')，就可以在 excel 中输出中文了。默认是 ascii。当然要记得在文件头部添加：

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

style\_compression 表示是否压缩，不常用。

Workbook 还有一些属性：

Owner 设置文档所有者。

country\_code: 国家码

wnd\_protect: 窗口保护

obj\_protect: 对象保护

Protect: 保护

backup\_on\_save: 保存时备份

Hpos: 横坐标

Vpos: 纵坐标

Width: 宽度

Height: 高度

active\_sheet: 活动 sheet

tab\_width: tab 宽度

wnd\_visible: 窗口是否可见

wnd\_mini: 窗口最小化

hscroll\_visible: 横向滚动条是否可见。

vscroll\_visible: 纵向滚动条是否可见。

tabs\_visible: tab 是否可见。

dates\_1904: 是否使用 1904 日期系统

use\_cell\_values: 单元格的值

default\_style: 默认样式

colour\_RGB: 颜色

比如设置国家码，代码：country.py:

```
from xlwt import *

w = Workbook()
w.country_code = 61
ws = w.add_sheet('AU')
w.save('country.xls')
```

方法有: add\_style, add\_font, add\_str, del\_str, str\_index, add\_rt, rt\_index, add\_sheet,

get\_sheet, raise\_bad\_sheetname, convert\_sheetindex, setup\_xcall, add\_sheet\_reference。

## 插入图片

**add\_sheet** 会返回一个 **Worksheet** 类。创建的时候有可选参数 **cell\_overwrite\_ok**, 表示是否可以覆盖单元格, 其实是 **Worksheet** 实例化的一个参数, 默认值是 **False**。

**Worksheet** 初始化的参数有 **sheetname**, **parent\_book**, **cell\_overwrite\_ok**。

**Worksheet** 的属性有: **Row**, **Column**, **explicit\_magn\_setting** (默认 **False**), **visibility** (默认 **0**), **split\_position\_units\_are\_twips** (默认 **False**), **row\_default\_height\_mismatch**, **row\_default\_hidden**, **row\_default\_space\_above**, **row\_default\_space\_below**, **last\_used\_row**, **first\_used\_row**, **last\_used\_col**, **row\_tempfile**。以上属性类定义中。

函数构成的属性有: **name**, **parent** (只读), **rows** (只读), **cols** (只读), **merged\_ranges** (只读), **bmp\_rec** (只读), **show\_formulas**, **show\_grid**, **show\_headers**, **panes\_frozen**, **auto\_colour\_grid**, **cols\_right\_to\_left**, **show\_outline**, **remove\_splits**, **selected**, **sheet\_visible**, **page\_preview**, **first\_visible\_row**, **first\_visible\_col**, **grid\_colour**, **preview\_magn**, **normal\_magn**, **scl\_magn**, **vert\_split\_pos**, **horz\_split\_pos**, **vert\_split\_first\_visible**, **horz\_split\_first\_visible**, **show\_auto\_page\_breaks**, **dialogue\_sheet**, **auto\_style\_outline**, **outline\_below**, **outline\_right**, **fit\_num\_pages**, **show\_row\_outline**, **show\_col\_outline**, **alt\_expr\_eval**, **alt\_formula\_entries**, **row\_default\_height**, **col\_default\_width**, **calc\_mode**, **calc\_count**, **RC\_ref\_mode**, **iterations\_on**, **delta**, **save\_recalc**, **print\_headers**, **print\_grid**, **vert\_page\_breaks**, **horz\_page\_breaks**, **header\_str**, **footer\_str**, **print\_centered\_vert**, **print\_centered\_horz**, **left\_margin**, **right\_margin**, **top\_margin**, **bottom\_margin**, **paper\_size\_code**, **print\_scaling**, **start\_page\_number**, **fit\_width\_to\_pages**, **fit\_height\_to\_pages**, **print\_in\_rows**, **portrait**, **print\_colour**, **print\_draft**, **print\_notes**, **print\_notes\_at\_end**, **print\_omit\_errors**, **print\_hres**, **print\_vres**, **header\_margin**, **footer\_margin**, **copies\_num**, **wnd\_protect**, **obj\_protect**, **protect**, **scen\_protect**, **password**。

方法有: **get\_parent**, **write**, **write\_rich\_text**, **merge**, **write\_merge**, **insert\_bitmap**, **col**, **row**, **row\_height**, **col\_width**。

下面例子使用 **insert\_bitmap** 来插入图片。代码: **image.py**:

```
from xlwt import *,

w = Workbook()
ws = w.add_sheet('Image')
ws.insert_bitmap('python.bmp', 2, 2)
ws.insert_bitmap('python.bmp', 10, 2)

w.save('image.xls')
```

## 设置样式

下面例子改变字体的高度。代码：row\_styles.py:

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-
# Copyright (C) 2005 Kiseliiov Roman

from xlwt import *

w = Workbook(encoding='utf-8')
ws = w.add_sheet('Hey, Dude')

for i in range(6, 80):
    fnt = Font()
    fnt.height = i*20
    style = XFStyle()
    style.font = fnt
    ws.write(i, 1, '武冈')
    ws.row(i).set_style(style)
w.save('row_styles.xls')
```

XFStyle 用于设置字体样式，有描述字符串 num\_format\_str，字体 font，居中 alignment，边界 borders，模式 pattern，保护 protection 等属性。另外还可以不写单元格，直接设置格式，比如代码 row\_styles\_empty:

```
from pyExcelerator import *

w = Workbook()
ws = w.add_sheet('Hey, Dude')

for i in range(6, 80):
    fnt = Font()
    fnt.height = i*20
    style = XFStyle()
    style.font = fnt
    ws.row(i).set_style(style)
w.save('row_styles_empty.xls')
```

设置列宽: `ws.col(i).width = 0x0d00 + i`

## 公式

`Formula` 方法可以生成公式, 注意 `Formula` 中的公式是不需要等号的。下面例子 `simple.py` 输出了红色的“Test”, 并在第 3 行包含了公式。

```
import xlwt
from datetime import datetime

font0 = xlwt.Font()
font0.name = 'Times New Roman'
font0.colour_index = 2
font0.bold = True

style0 = xlwt.XFStyle()
style0.font = font0

style1 = xlwt.XFStyle()
style1.num_format_str = 'D-MMM-YY'

wb = xlwt.Workbook()
ws = wb.add_sheet('A Test Sheet')

ws.write(0, 0, 'Test', style0)
ws.write(1, 0, datetime.now(), style1)
ws.write(2, 0, 1)
ws.write(2, 1, 1)
ws.write(2, 2, xlwt.Formula("A3+B3"))

wb.save('example.xls')
```

这里另有一个公式的实例 `parse-fmla.py`:

```
from xlwt import ExcelFormulaParser, ExcelFormula
import sys

f = ExcelFormula.Formula(
    """ -(1.80 + 2.898 * 1)/(1.80 + 2.898)*
    AVERAGE((1.80 + 2.898 * 1)/(1.80 + 2.898);
             (1.80 + 2.898 * 1)/(1.80 + 2.898);
             (1.80 + 2.898 * 1)/(1.80 + 2.898)) +
```

```
SIN(PI()/4) """)
```

## 合并单元格

`write_merge` 可以合并单元格, 注意 1,2 个参数表示行数, 3,4 的参数表示列数。实例: `merged0.py`。

```
from xlwt import *

wb = Workbook()
ws0 = wb.add_sheet('sheet0')

fnt = Font()
fnt.name = 'Arial'
fnt.colour_index = 4
fnt.bold = True

borders = Borders()
borders.left = 6
borders.right = 6
borders.top = 6
borders.bottom = 6

style = XFStyle()
style.font = fnt
style.borders = borders

ws0.write_merge(3, 3, 1, 5, 'test1', style)
ws0.write_merge(4, 10, 1, 5, 'test2', style)
ws0.col(1).width = 0x0d00

wb.save('merged0.xls')
```

## 日期格式

`XFStyle` 类的 `num_format_str` 属性可以设置数值的输出格式, 也对日期生效。日期格式的实例, 代码: `dates.py`

```
from xlwt import *
from datetime import datetime

w = Workbook()
ws = w.add_sheet('Hey, Dude')

fmts = [
```



```

'M/D/YY',
'D-MMM-YY',
'D-MMM',
'MMM-YY',
'h:mm AM/PM',
'h:mm:ss AM/PM',
'h:mm',
'h:mm:ss',
'M/D/YY h:mm',
'mm:ss',
'[h]:mm:ss',
'mm:ss.0',
]

i = 0
for fmt in fmts:
    ws.write(i, 0, fmt)

    style = XFStyle()
    style.num_format_str = fmt

    ws.write(i, 4, datetime.now(), style)

    i += 1

w.save('dates.xls')

```

## 边框

Borders 类的 left, right, bottom, top 属性分别可以设置左右低高的边框, Font 类的 name 可以设置字体类型, struck\_out 为是否添加删除线, bold 为是否为粗体, 下面展示了不同边框和删除样式的字体, 代码 blanks.py:

```

from xlwt import *

font0 = Font()
font0.name = 'Times New Roman'
font0.struck_out = True
font0.bold = True

style0 = XFStyle()
style0.font = font0

wb = Workbook()

```

```

ws0 = wb.add_sheet('0')

ws0.write(1, 1, 'Test', style0)

for i in range(0, 0x53):
    borders = Borders()
    borders.left = i
    borders.right = i
    borders.top = i
    borders.bottom = i

    style = XFStyle()
    style.borders = borders

    ws0.write(i, 2, '', style)
    ws0.write(i, 3, hex(i), style0)

ws0.write_merge(5, 8, 6, 10, "")

wb.save('blanks.xls')

```

## 字体颜色

Font 中的 `colour_index` 可以设置颜色，下面展示了不同颜色的字体，可以作为颜色参考，代码 `format.py`:

```

from xlwt import *

font0 = Font()
font0.name = 'Times New Roman'
font0.struck_out = True
font0.bold = True

style0 = XFStyle()
style0.font = font0

wb = Workbook()
ws0 = wb.add_sheet('0')

ws0.write(1, 1, 'Test', style0)

for i in range(0, 0x53):

```

```

fnt = Font()
fnt.name = 'Arial'
fnt.colour_index = i
fnt.outline = True

borders = Borders()
borders.left = i

style = XFStyle()
style.font = fnt
style.borders = borders

ws0.write(i, 2, 'colour', style)
ws0.write(i, 3, hex(i), style0)

wb.save('format.xls')

```

## 超级链接

Formula 中的可以插入 HYPERLINK 超级链接，代码 hyperlinks.py:

```

from xlwt import *

f = Font()
f.height = 20*72
f.name = 'Verdana'
f.bold = True
f.underline = Font.UNDERLINE_DOUBLE
f.colour_index = 4

h_style = XFStyle()
h_style.font = f

w = Workbook()
ws = w.add_sheet('F')

#####
## NOTE: parameters are separated by semicolon!!!
#####

n = "HYPERLINK"
ws.write_merge(1, 1, 1, 10, Formula(n +
'"http://www.irs.gov/pub/irs-pdf/f1000.pdf";"f1000.pdf"'), h_style)

```

```

ws.write_merge(2, 2, 2, 25, Formula(n +
    ' ("mailto:roman.kiseliov@gmail.com?subject=pyExcelerator-feedback&Body=Hello,%20Roman!";"pyExcelerator-feedback")'), h_style)

w.save("hyperlinks.xls")

```

## 编码

在没有指定编码的情况下，也可以通过 `unicode` 输出字符，不过这样比较费劲，建议还是使用 `utf-8` 编码，代码 `unicode1.py`：

```

from xlwt import *

w = Workbook()
ws1 = w.add_sheet(u'\N{GREEK SMALL LETTER ALPHA}\N{GREEK SMALL LETTER BETA}\N{GREEK SMALL LETTER GAMMA}')

ws1.write(0, 0, u'\N{GREEK SMALL LETTER ALPHA}\N{GREEK SMALL LETTER BETA}\N{GREEK SMALL LETTER GAMMA}')
ws1.write(1, 1, u'\N{GREEK SMALL LETTER DELTA}x = 1 + \N{GREEK SMALL LETTER DELTA}')

ws1.write(2,0, u'A\u2262\u0391.') # RFC2152 example
ws1.write(3,0, u'Hi Mom -\u263a-!') # RFC2152 example
ws1.write(4,0, u'\u65E5\u672C\u8A9E') # RFC2152 example
ws1.write(5,0, u'Item 3 is \u00a31.') # RFC2152 example
ws1.write(8,0, u'\N{INTEGRAL}') # RFC2152 example

w.add_sheet(u'A\u2262\u0391.') # RFC2152 example
w.add_sheet(u'Hi Mom -\u263a-!') # RFC2152 example
one_more_ws = w.add_sheet(u'\u65E5\u672C\u8A9E') # RFC2152 example
w.add_sheet(u'Item 3 is \u00a31.') # RFC2152 example

one_more_ws.write(0, 0, u'\u2665\u2665')

w.add_sheet(u'\N{GREEK SMALL LETTER ETA WITH TONOS}')
w.save('unicode1.xls')

```

## 冻结

冻结设置 `panes_frozen` 为 `True`，然后设置冻结的位置就好。支持行冻结，列冻结及相关的隐藏功能。代码：`panes.py`：

```

from xlwt import *

w = Workbook()

```

```
ws1 = w.add_sheet('sheet 1')
ws2 = w.add_sheet('sheet 2')
ws3 = w.add_sheet('sheet 3')
ws4 = w.add_sheet('sheet 4')
ws5 = w.add_sheet('sheet 5')
ws6 = w.add_sheet('sheet 6')

for i in range(0x100):
    ws1.write(i/0x10, i%0x10, i)

for i in range(0x100):
    ws2.write(i/0x10, i%0x10, i)

for i in range(0x100):
    ws3.write(i/0x10, i%0x10, i)

for i in range(0x100):
    ws4.write(i/0x10, i%0x10, i)

for i in range(0x100):
    ws5.write(i/0x10, i%0x10, i)

for i in range(0x100):
    ws6.write(i/0x10, i%0x10, i)

ws1.panes_frozen = True
ws1.horz_split_pos = 2

ws2.panes_frozen = True
ws2.vert_split_pos = 2

ws3.panes_frozen = True
ws3.horz_split_pos = 1
ws3.vert_split_pos = 1

ws4.panes_frozen = False
ws4.horz_split_pos = 12
ws4.horz_split_first_visible = 2

ws5.panes_frozen = False
ws5.vert_split_pos = 40
ws4.vert_split_first_visible = 2

ws6.panes_frozen = False
```

```

ws6.horz_split_pos = 12
ws4.horz_split_first_visible = 2
ws6.vert_split_pos = 40
ws4.vert_split_first_visible = 2

w.save('panes.xls')

```

## 数值格式

和日期格式类似，代码：num\_formats.py:

```

from xlwt import *

w = Workbook()
ws = w.add_sheet('Hey, Dude')

fmts = [
    'general',
    '0',
    '0.00',
    '#,##0',
    '#,##0.00',
    '"$"#,##0_);("$"#,##',
    '"$"#,##0_);[Red]("$"#,##',
    '"$"#,##0.00_);("$"#,##',
    '"$"#,##0.00_);[Red]("$"#,##',
    '0%',
    '0.00%',
    '0.00E+00',
    '# ?/?',
    '# ??/??',
    'M/D/YY',
    'D-MMM-YY',
    'D-MMM',
    'MMM-YY',
    'h:mm AM/PM',
    'h:mm:ss AM/PM',
    'h:mm',
    'h:mm:ss',
    'M/D/YY h:mm',
    '_(##,##0_);(##,##0)',
    '_(##,##0_);[Red](##,##0)',
    '_(##,##0.00_);(##,##0.00)',
    '_(##,##0.00_);[Red](##,##0.00)',
    '_("$"* ##,##0_);_("$"* (##,##0);_("$"* "-"_);_(@_)',

```

```

'_( * #,##0_);_( * ( #,##0);_( * "-"_);_( @_ )',
'_( "$" * #,##0.00_);_( "$" * ( #,##0.00);_( "$" * "-"??_);_( @_ )',
'_( * #,##0.00_);_( * ( #,##0.00);_( * "-"??_);_( @_ )',
'mm:ss',
'[h]:mm:ss',
'mm:ss.0',
'##0.0E+0',
'@'
]

i = 0
for fmt in fmts:
    ws.write(i, 0, fmt)

    style = XFStyle()
    style.num_format_str = fmt

    ws.write(i, 4, -1278.9078, style)

    i += 1

w.save('num_formats.xls')

```

## 更多公式

Formulas.py 有更多公式可供参考:

```

from xlwt import *

w = Workbook()
ws = w.add_sheet('F')

ws.write(0, 0, Formula("(1+1)"))
ws.write(1, 0, Formula("(1+1)/(-2-2)"))
ws.write(2, 0, Formula("(134.8780789+1)"))
ws.write(3, 0, Formula("(134.8780789e-10+1)"))
ws.write(4, 0, Formula("-1/(1+1)+9344"))

ws.write(0, 1, Formula("(1+1)"))
ws.write(1, 1, Formula("(1+1)/(-2-2)"))
ws.write(2, 1, Formula("(134.8780789+1)"))
ws.write(3, 1, Formula("(134.8780789e-10+1)"))
ws.write(4, 1, Formula("-1/(1+1)+9344"))

ws.write(0, 2, Formula("A1*B1"))

```

```

ws.write(1, 2, Formula("A2*B2"))
ws.write(2, 2, Formula("A3*B3"))
ws.write(3, 2, Formula("A4*B4*sin(pi()/4)"))
ws.write(4, 2, Formula("A5*B5*pi()/1000"))

#####
## NOTE: parameters are separated by semicolon!!!
#####

ws.write(5, 2,
Formula("C1+C2+C3+C4+C5/(C1+C2+C3+C4/(C1+C2+C3+C4/(C1+C2+C3+C4)+C5)+C5)-20.3e-2
"))
ws.write(5, 3, Formula("C1^2"))
ws.write(6, 2, Formula("SUM(C1;C2;;;C3;;;C4)"))
ws.write(6, 3, Formula("SUM($A$1:$C$5)"))

ws.write(7, 0, Formula("lkjljllkllk1"))
ws.write(7, 1, Formula("yuyiyiyiyi"))
ws.write(7, 2, Formula("A8 & B8 & A8"))
ws.write(8, 2, Formula("now()"))

ws.write(10, 2, Formula("TRUE"))
ws.write(11, 2, Formula("FALSE"))
ws.write(12, 3, Formula("IF(A1>A2;3;"hkjhjkhk"))))

w.save('formulas.xls')

```

## 保护

```

from xlwt import *

fnt = Font()
fnt.name = 'Arial'
fnt.colour_index = 4
fnt.bold = True

borders = Borders()
borders.left = 6
borders.right = 6
borders.top = 6
borders.bottom = 6

style = XFStyle()
style.font = fnt

```



```

style.borders = borders

wb = Workbook()

ws0 = wb.add_sheet('Rows Outline')

ws0.write_merge(1, 1, 1, 5, 'test 1', style)
ws0.write_merge(2, 2, 1, 4, 'test 1', style)
ws0.write_merge(3, 3, 1, 3, 'test 2', style)
ws0.write_merge(4, 4, 1, 4, 'test 1', style)
ws0.write_merge(5, 5, 1, 4, 'test 3', style)
ws0.write_merge(6, 6, 1, 5, 'test 1', style)
ws0.write_merge(7, 7, 1, 5, 'test 4', style)
ws0.write_merge(8, 8, 1, 4, 'test 1', style)
ws0.write_merge(9, 9, 1, 3, 'test 5', style)

ws0.row(1).level = 1
ws0.row(2).level = 1
ws0.row(3).level = 2
ws0.row(4).level = 2
ws0.row(5).level = 2
ws0.row(6).level = 2
ws0.row(7).level = 2
ws0.row(8).level = 1
ws0.row(9).level = 1

ws1 = wb.add_sheet('Columns Outline')

ws1.write_merge(1, 1, 1, 5, 'test 1', style)
ws1.write_merge(2, 2, 1, 4, 'test 1', style)
ws1.write_merge(3, 3, 1, 3, 'test 2', style)
ws1.write_merge(4, 4, 1, 4, 'test 1', style)
ws1.write_merge(5, 5, 1, 4, 'test 3', style)
ws1.write_merge(6, 6, 1, 5, 'test 1', style)
ws1.write_merge(7, 7, 1, 5, 'test 4', style)
ws1.write_merge(8, 8, 1, 4, 'test 1', style)
ws1.write_merge(9, 9, 1, 3, 'test 5', style)

ws1.col(1).level = 1
ws1.col(2).level = 1
ws1.col(3).level = 2
ws1.col(4).level = 2
ws1.col(5).level = 2

```

```
ws1.col(6).level = 2
ws1.col(7).level = 2
ws1.col(8).level = 1
ws1.col(9).level = 1

ws2 = wb.add_sheet('Rows and Columns Outline')

ws2.write_merge(1, 1, 1, 5, 'test 1', style)
ws2.write_merge(2, 2, 1, 4, 'test 1', style)
ws2.write_merge(3, 3, 1, 3, 'test 2', style)
ws2.write_merge(4, 4, 1, 4, 'test 1', style)
ws2.write_merge(5, 5, 1, 4, 'test 3', style)
ws2.write_merge(6, 6, 1, 5, 'test 1', style)
ws2.write_merge(7, 7, 1, 5, 'test 4', style)
ws2.write_merge(8, 8, 1, 4, 'test 1', style)
ws2.write_merge(9, 9, 1, 3, 'test 5', style)

ws2.row(1).level = 1
ws2.row(2).level = 1
ws2.row(3).level = 2
ws2.row(4).level = 2
ws2.row(5).level = 2
ws2.row(6).level = 2
ws2.row(7).level = 2
ws2.row(8).level = 1
ws2.row(9).level = 1

ws2.col(1).level = 1
ws2.col(2).level = 1
ws2.col(3).level = 2
ws2.col(4).level = 2
ws2.col(5).level = 2
ws2.col(6).level = 2
ws2.col(7).level = 2
ws2.col(8).level = 1
ws2.col(9).level = 1

ws0.protect = True
ws0.wnd_protect = True
ws0.obj_protect = True
ws0.scen_protect = True
ws0.password = "123456"
```

```

ws1.protect = True
ws1.wnd_protect = True
ws1.obj_protect = True
ws1.scen_protect = True
ws1.password = "abcdefghij"

ws2.protect = True
ws2.wnd_protect = True
ws2.obj_protect = True
ws2.scen_protect = True
ws2.password = "ok"

wb.protect = True
wb.wnd_protect = True
wb.obj_protect = True
wb.save('protection.xls')

```

## 综合实例

下面程序 `xlwt_easyxf_simple_demo.py` 的 `write_xls` 对输出 `xls` 进行了一定封装, 比较有实用意义。

```

import xlwt
import datetime
ezxf = xlwt.easyxf

def write_xls(file_name, sheet_name, headings, data, heading_xf, data_xfs):
    book = xlwt.Workbook()
    sheet = book.add_sheet(sheet_name)
    rowx = 0
    for colx, value in enumerate(headings):
        sheet.write(rowx, colx, value, heading_xf)
    sheet.set_panes_frozen(True) # frozen headings instead of split panes
    sheet.set_horz_split_pos(rowx+1) # in general, freeze after last heading row
    sheet.set_remove_splits(True) # if user does unfreeze, don't leave a split
    there
    for row in data:
        rowx += 1
        for colx, value in enumerate(row):
            sheet.write(rowx, colx, value, data_xfs[colx])
    book.save(file_name)

if __name__ == '__main__':
    import sys
    mkd = datetime.date

```

```

hdngs = ['Date', 'Stock Code', 'Quantity', 'Unit Price', 'Value', 'Message']
kinds = 'date    text          int          price          money
text'.split()
data = [
    [mkd(2007, 7, 1), 'ABC', 1000, 1.234567, 1234.57, ''],
    [mkd(2007, 12, 31), 'XYZ', -100, 4.654321, -465.43, 'Goods returned'],
] + [
    [mkd(2008, 6, 30), 'PQRCD', 100, 2.345678, 234.57, ''],
] * 100

heading_xf = ezxf('font: bold on; align: wrap on, vert centre, horiz center')
kind_to_xf_map = {
    'date': ezxf(num_format_str='yyyy-mm-dd'),
    'int': ezxf(num_format_str='#,##0'),
    'money': ezxf('font: italic on; pattern: pattern solid, fore-colour
grey25',
    num_format_str='$#,##0.00'),
    'price': ezxf(num_format_str='#0.000000'),
    'text': ezxf(),
}
data_xfs = [kind_to_xf_map[k] for k in kinds]
write_xls('xlwt_easyxf_simple_demo.xls', 'Demo', hdngs, data, heading_xf,
data_xfs)

```