

API参考

class `xlwt.Workbook.Workbook` (*encoding='ascii', style_compression = 0*)

这是一个表示工作簿及其所有内容的类。使用xlwt创建Excel文件时，通常首先要实例化此类的对象。

add_sheet (*sheetname, cell_overwrite_ok = False*)

此方法用于在工作簿中创建工作表。

- 参数:**
- **sheetname** - 用于此工作表的名称，因为它将显示在Excel应用程序底部的选项卡中。
 - **cell_overwrite_ok** - 如果 `True` 写入多次，则添加的工作表中的单元格不会引发异常。

返回: 在 `Worksheet` 已添加。

save (*filename_or_stream*)

此方法用于将工作簿保存为本机Excel格式的文件。

- 参数:** **filename_or_stream** - 这可以是包含文件文件名的字符串，在这种情况下，使用提供的名称将excel文件保存到磁盘。它也可以是具有write方法的流对象，例如 a `StringIO`，在这种情况下，excel文件的数据被写入流。

class `xlwt.Worksheet.Worksheet` (*sheetname, parent_book, cell_overwrite_ok = False*)

这是一个表示工作簿中工作表内容的类。

! 警告

您通常不会自己创建此类的实例。他们从打电话回来 `add_sheet()`。

write (*r, c, label="", style = <xlwt.Style.XFStyle object>*)

此方法用于将单元格写入a `Worksheet`。

参数:

- `r` - 应写入单元格的工作表中行的零相对数。
- `c` - 工作表中应写入单元格的列的零相对数。
- **标签** -
要写入的数据值。

An `int` , `long` 或 `Decimal` 实例转换为 `float` 。

一个 `unicode` 实例写成的。甲 `bytes` 实例转换到 `unicode` 使用的编码, 缺省 `ascii` 时, 当指定的 `Workbook` 被创建的实例。

甲 `datetime` , `date` 或 `time` 实例转换到Excel日期格式 (表示自 (通常) 的天数的浮子 `1899-12-31T00:00:00` , 假装1900是闰年下) 。

一个 `bool` 实例将显示为 `TRUE` 或 `FALSE` Excel中。

`None` 导致单元格为空: 没有数据, 只有格式。

一个 `xlwt.Formula` 实例将导致要被写入一个Excel公式。

- **风格** -
样式 (也称为XF (扩展格式)) 是一个 `XFStyle` 对象, 它封装应用于单元格及其内容的格式。
`XFStyle` 最好使用该 `easyxf()` 功能设置对象 。它们也可以通过在设置属性设置 `Alignment` , `Borders` , `Pattern` , `Font` 和 `Protection` 对象然后设置这些对象和一个格式字符串作为属性 `XFStyle` 对象。

格式化

XF记录能够存储显式单元格格式属性或单元格样式的属性。显式格式包括对单元格样式XF记录的引用。这允许使用一些显式属性扩展定义的单元格样式。格式属性分为6组:

组	属性
数字格式	数字格式索引 (FORMAT记录的索引)
字形	字体索引 (FONT记录的索引)
对准	水平和垂直对齐, 文本换行, 缩进, 方向/旋转, 文本方向
边界	边框线条样式和颜色
背景	背景区域样式和颜色
保护	细胞锁定, 公式隐藏

对于每个组, 单元格XF记录中的标志指定是使用该XF记录中包含的属性还是引用的样式XF记录中的属性。在样式XF记录中, 这些标志指定在将样式应用于单元格时属性是否将覆盖显式单元格格式。更改单元格样式 (不将此样式应用于单元格) 将更改已使用该样式的所有单元格, 并且不包含已更改样式属性的显式单元格属性。如果单元格XF记录不包含组中的显式属性 (如果未设置属性组标志) , 则它会重复其样式XF记录的属性。

```
xlwt.Style.easyxf (strg_to_parse =", num_format_str = None, field_sep =', ', line_sep =';',  
intro_sep =': ', esc_char = '\\', debug = False )
```

此函数用于创建和配置 `XfStyle` 与（例如）`Worksheet.write()` 方法一起使用的对象。

它需要一个要解析的字符串，以获得属性值 `Alignment` , `Borders` , `Font` , `Pattern` 和 `Protection` 对象。

请参阅文件 `examples / xlwt_easyxf_simple_demo.py` 中的示例和 `xf_dict` 字典中的示例 `xlwt.Style` 。

允许各种同义词，包括颜色/颜色，中心/中心和灰色/灰色。案例无关紧要（除了字体名称外）。 `-` 可以用来代替 `_` 。

例： `font: bold on; align: wrap on, vert centre, horiz center`

参数： `num_format_str` _

要获取要重现其格式的现有单元格的“数字格式字符串”，请选择单元格并单击“格式/单元格/数字/自定义”。否则，请参阅Excel帮助。

例如： `"#,##0.00"` , `"dd/mm/yyyy"`

返回： 一个 `XfStyle` 对象。