

API参考

xlrd

```
xlrd.open_workbook (filename = None, logfile = <_io.TextIOWrapper name = '<stdout>' mode = 'w' coding = 'UTF-8'>, verbosity = 0, use_mmap = 1, file_contents = None, encoding_override = None, formatting_info = False, on_demand = False, ragged_rows = False )
```

打开电子表格文件以进行数据提取。

参数:

- filename - 要打开的电子表格文件的路径。
- logfile - 写入消息和诊断的打开文件。
- 详细程度 - 增加写入日志文件的跟踪材料的数量。
- use_mmap -
是否使用mmap模块是启发式确定的。使用此arg覆盖结果。
当前启发式: 如果存在, 则使用mmap。
- file_contents - 字符串或 `mmap.mmap` 对象或其他类似行为的对象。如果 `file_contents` 提供, `filename` 将不会使用, 除了 (可能) 在消息中。
- encoding_override - 用于克服旧版本文件中丢失或错误的代码页信息。请参阅处理 Unicode。
- formatting_info -
默认值为 `False`, 可以节省内存。在这种情况下, 通过忽略文件 `BLANK` 和 `MULBLANK` 记录, “空白”单元格 (具有自己的格式信息但没有数据的单元格) 将被视为空。这会切断空单元格或空白单元格的任何底部或右边“边距”。只有 `cell_value()` 并且 `cell_type()` 可用。
何时 `True`, 将从电子表格文件中读取格式信息。这提供了所有细胞, 包括空细胞和空细胞。格式化信息可用于每个单元格。
请注意, 当与xlsx文件一起使用时, 这将引发 `NotImplementedError`。
- on_demand - 管理表单最初是否全部加载或者是否由调用者请求。请参阅按需加载工作表。
- ragged_rows -
默认值 `False` 表示所有行都使用空单元格填充, 以便所有行具有与中找到的相同的大小 `ncols`。
`True` 意味着行的末尾没有空单元格。如果行的大小变化很大, 这可以节省大量内存。另见 `row_len()` 方法。

返回:

`Book` 类的一个实例。

```
xlrd.dump (filename, outfile = <_io.TextIOWrapper name = '<stdout>' mode = 'w' coding = 'UTF-8'>, unnumbered = False )
```

对于调试: 在char和hex中转储XLS文件的BIFF记录。

- 参数:
- filename - 要转储的文件的路径。
 - outfile - 打开转储的文件。
 - unnumbered - 如果为true, 则省略偏移量 (对于有意义的差异)。

`xlrd.count_records (filename, outfile = <_io.TextIOWrapper name='<stdout>' mode='w' coding='UTF-8'>)`

对于调试和分析: 汇总文件的BIFF记录。即: 生成一个排序文件。 `(record_name, count)`

- 参数:
- filename - 要汇总的文件的路径。
 - outfile - 打开摘要的打开文件。

xlrd.biffh

异常 `xlrd.biffh.XLRDError`

指示从Excel文件读取数据时出现问题的异常。

类 `xlrd.biffh.BaseObject`

包中几乎所有其他类的父类。定义 `dump()` 调试的常用方法。

`dump (f=无, 标题=无, 页脚=无, 缩进=0)`

- 参数:
- f - 打开转储的文件对象
 - header - 转储前要写入的文本
 - 页脚 - 转储后要写入的文本
 - indent - 前导空格数 (用于递归调用)

`xlrd.biffh.error_text_from_code`
`= {0: '#NULL!', 7: '#DIV/0!', 15: '#VALUE!', 23: '#REF!', 29: '#NAME?', 36: '#NUM!', 42: '#N/A'}`

此字典可用于生成Excel用于错误单元格的内部代码的文本版本。

`xlrd.biffh.unpack_unicode (data, pos, lenlen = 2)`

返回unicode_strg

`xlrd.biffh.unpack_unicode_update_pos (data, pos, lenlen = 2, known_len = None)`

返回 (unicode_strg, pos的更新值)

xlrd.book

类 `xlrd.book.Name`

与命名参考, 公式, 宏等有关的信息

❗ 注意

名称信息**不是**从早于Excel 5.0 () 的文件中提取的 `Book.biff_version < 50`

hidden=

0 =可见; 1 =隐藏

func=

0 =命令宏; 1 =功能宏。仅在宏== 1时相关

vbasic=

0 =表格宏; 1 = VisualBasic宏。仅在宏== 1时相关

macro=

0 =标准名称; 1 =宏名称

complex=

0 =简单公式; 1 =复杂公式（数组公式或用户定义）。

❗ 注意

没有看到任何例子。

builtin=

0 =用户定义的名称; 1 =内置名称

常见例子: `Print_Area` , `Print_Titles` ; 请参阅OOo文档以获取完整列表

funcgroup=

功能组。仅在宏== 1时相关; 请参阅OOo docs for values。

binary=

0 =公式定义; 1 =二进制数据

❗ 注意

没有看到任何例子。

name_index=

book.name_obj_list中此对象的索引

raw_formula=

一个8位字符串。

scope=

-1 :

名称是全局的（在所有计算表中都可见）。

-2 :

该名称属于宏表或VBA表。

-3 :

该名称无效。

0 <= scope < book.nsheets :

该名称是索引为范围的工作表的本地名称。

cell ()

对于频繁使用情况，这是一种便利方法，其中名称是指单个单元格。

返回： `Cell` 类的一个实例。

举： `xlrd.biffh.XLRDError` - 名称不是对单个单元格的常量绝对引用。

area2d (clipped = True)

这是一种用例的便捷方法，其中名称是指一个工作表中的一个矩形区域。

参数： `clipped` - 如果 `True` 是默认值，则剪切返回的矩形以适应。保证该区域中可用行的数量（可能为零）；同样用于柱

子。 `(0, sheet.nrows, 0, sheet.ncols)` `0 <= rowxlo <= rowxhi <= sheet.nrows` `rowxhi - rowxlo`

返回： 一个元组。 `(sheet_object, rowxlo, rowxhi, colxlo, colxhi)`

举： `xlrd.biffh.XLRDError` - 名称不是对单个工作表中单个区域的常量绝对引用。

类 `xlrd.book.Book`

“工作簿”的内容。

警告

你不应该自己实例化这个类。您使用 `Book` 调用时返回的对象 `open_workbook()`。

datemode=

上次保存此文件时，哪个日期系统生效。

0:

1900系统（Excel默认为Excel）。

1:

1904系统（默认为Excel for Macintosh）。

如果未在文件中指定，则默认为0。

biff_version=

用于创建文件的BIFF（二进制交换文件格式）版本。最新版本是8.0（此处表示为80），随Excel 97引入。此模块最早支持：2.0（表示为20）。

codepage=无

一个整数，表示用于此文件中字符串的字符集。对于BIFF 8及更高版本，这将是1200，意味着Unicode；更确切地说，UTF_16_LE。对于早期版本，这用于派生用于转换为Unicode的适当Python编码。例如：， `1252 -> 'cp1252'` `10000 -> 'mac_roman'`

encoding=无

从代码页派生的编码。

countries= (0,0)

包含电话国家/地区代码的元组：

`[0]`：

创建文件时的用户界面设置。

`[1]`：

区域设置。

示例：含义。 `(1, 61)` `(USA, Australia)`

此信息可能为未知代码页的正确编码提供线索。有关观察值的长列表，请参阅OpenOffice.org文档以获取 `COUNTRY` 记录。

user_name=

记录什么（如果有的话）作为保存文件的最后一个用户的名称。

font_list= []

`Font` 类实例列表，每个实例对应一个FONT记录。

版本0.6.1中的新功能。

format_list= []

一个 `Format` 对象列表，每个对象对应一条 `FORMAT` 记录，按它们在输入文件中出现的顺序排列。它不包含内置格式。

如果要使用（例如）创建输出文件 `xlwt`，请使用此列表。

用于所有视觉渲染目的的集合是 `format_map`。

版本0.6.1中的新功能。

format_map= {}

从映射 `format_key` 到 `Format` 对象。

版本0.6.1中的新功能。

load_time_stage_1=

将XLS图像提取为连续字符串（或等效的mmap）的时间（以秒为单位）。

load_time_stage_2=

从连续字符串（或等效的mmap）解析数据的时间（以秒为单位）。

sheets ()

返回： 书中所有表格的列表。

将加载所有尚未加载的纸张。

sheet_by_index (sheetx)

参数： sheetx - 工作表索引 `range(nsheets)`

返回： 一 `Sheet` 。

sheet_by_name (sheet_name)

参数： sheet_name - 所需工作表的名称。

返回： 一 `Sheet` 。

sheet_names ()

返回： 工作簿文件中所有工作表的名称列表。即使尚未装入纸张，也可以使用此信息。

sheet_loaded (sheet_name_or_index)

参数： sheet_name_or_index - 查询的工作表的名称或索引

返回： `True` 如果已加载工作表， `False` 否则。

版本0.7.1中的新功能。

unload_sheet (sheet_name_or_index)

参数： sheet_name_or_index - 要卸载的工作表的名称或索引。

版本0.7.1中的新功能。

release_resources ()

这种方法有双重目的。 `mmap.mmap` 在 `on_demand` 模式下加载完工作表后，可以调用它来释放内存消耗对象和（可能）内存映射文件（对象），但仍然需要 `Book` 对象检查已加载的工作表。它也会自动调用（a）在 `open_workbook()` 引发异常时调用，（b）如果使用 `with` 语句， `with` 则在退出块时调用。在同一个对象上多次调用此方法没有任何不良影响。

`name_and_scope_map= {}`

从单个映射到单个映射 `(lower_case_name, scope)` `Name`

宾语。

版本0.6.0中的新功能。

`name_map= {}`

从 `lower_case_name` 到 `Name` 对象列表的映射。列表按范围顺序排序。通常，列表中将有有一个项目（全局范围）。

版本0.6.0中的新功能。

`nsheets=`

工作簿文件中存在的工作表数。即使尚未装入纸张，也可以使用此信息。

`name_obj_list= []`

包含工作簿中 `Name` 每个 `NAME` 记录的对象的列表。

版本0.6.0中的新功能。

`colour_map= {}`

这提供了颜色索引的定义。请参考 [调色板; 颜色索引](#)，用于说明如何在Excel中表示颜色。

颜色索引到调色板映射到元组。“魔术”索引例如映射到。 `(red, green, blue)` `0x7FFF` `None`

`colour_map` 如果要在屏幕上或PDF文件中渲染单元格，则需要它。如果您正在编写输出XLS文件，请使用 `palette_record`。

❗ 注意

仅在提取时提取 `open_workbook(..., formatting_info=True)`

版本0.6.1中的新功能。

`palette_record= []`

如果用户更改了标准调色板中的任何颜色，则XLS文件将包含其中包含 `PALETTE` 56（Excel 4.0及更早版本为16）RGB值的记录，此列表将为例如。否则此列表将为空。如果您正在编写输出XLS文件，那么这就是您所需要的。如果要在屏幕或PDF文件中渲染单元格，请使用。

`[(r0, b0, g0), ..., (r55, b55, g55)]` `colour_map`

❗ 注意

仅在提取时提取 `open_workbook(..., formatting_info=True)`

版本0.6.1中的新功能。

`xf_list= []`

`XF` 类实例列表，每个实例对应一条 `XF` 记录。

版本0.6.1中的新功能。

`style_name_map= {}`

这提供了通过名称访问内置样式和用户定义样式的扩展格式信息。

它映射 `name` 到，其中 可以是用户定义样式的名称，也可以是其中一个内置样式的名称。已知的内置名称是Normal，RowLevel_1到RowLevel_7，ColLevel_1到ColLevel_7，逗号，货币，百分比，“逗号[0]”，“货币[0]”，超链接和“跟随超链接”。 `(built_in, xf_index) name`

`built_in` 具有以下含义

1:

内置风格

0:

用户自定义

`xf_index` 是一个索引 `Book.xf_list` 。

参考文献：OOo docs s6.99 (`STYLE` 记录)；Excel UI格式/样式

版本0.6.1中的新功能。

仅在提取时提取 `open_workbook(..., formatting_info=True)`

版本0.7.4中的新功能。

`xlrd.book.unpack_SST_table (datatab, nstrings)`

返回字符串列表

xlrd.compdoc

实现从OLE2复合文档文件中提取“工作簿”或“工作簿”流（作为一个大字符串）所需的最小功能。

`xlrd.compdoc.SIGNATURE= B '\ XD0 \ XCF \ X11 \ xe0 \ XA1 \ XB1 \ X1A \ XE1'`

魔法cookie应该出现在文件的前8个字节中。

异常 `xlrd.compdoc.CompDocError`

`class xlrd.compdoc.CompDoc (mem, logfile = <_io.TextIOWrapper name = '<stdout>' mode = 'w' coding`

= 'UTF-8'>, DEBUG = 0)

复合文档处理程序

参数: mem - 文件的原始内容, 字符串或 `mmap.mmap` 对象。它需要支持的唯一操作是切片。

get_named_stream (qname)

询问复合文档的目录; 如果找到则返回流作为字符串, 否则返回 `None`。

参数: qname - 所需流的名称, 例如 `'Workbook'`。应该是Unicode或可转换为Unicode。

locate_named_stream (qname)

询问复合文档的目录。

如果找不到指定的流, 将返回。 `(None, 0, 0)`

如果找到指定的流并且在 `mem` 打开文档时使用的原始字节序列 () 内是连续的, 则返回。 `(mem, offset_to_start_of_stream, length_of_stream)`

否则, 将从片段构建新字符串并 返回。 `(new_string, 0, length_of_stream)`

参数: qname - 所需流的名称, 例如 `'Workbook'`。应该是Unicode或可转换为Unicode。

xlrd.formatting

格式化信息的模块。

xlrd.formatting.nearest_colour_index (colour_map, rgb, debug = 0)

通用功能。使用欧几里德距离。到目前为止仅用于前BIFF8 `WINDOW2` 记录。不一定要快。不一定要花哨。

类 xlrd.formatting.EqNeAttrs

这种混合类只存在使 `Format`, `Font` 和 `XF` 对象可通过其属性值进行比较。

类 xlrd.formatting.Font

Excel“字体”不仅包含通常被视为字体的内容的细节, 还包含其他几个显示属性的详细信息。项目对应于Excel UI格式 - >单元格 - >字体选项卡中的项目。

版本0.6.1中的新功能。

bold=

1 = 字符是粗体。多余的; 看“重量”属性。

character_set=

价值观:

0 = ANSI拉丁语1 =系统默认值2 =符号, 77 = Apple Roman, 128 = ANSI日语Shift-JIS, 129 = ANSI韩语 (韩语), 130 = ANSI韩语 (Johab), 134 = ANSI中文简体GBK, 136 = ANSI中文繁体BIG5, 161 = ANSI希腊语, 162 = ANSI土耳其语, 163 = ANSI越南语, 177 = ANSI希伯来语, 178 = ANSI阿拉伯语, 186 = ANSI波罗的海, 204 = ANSI西里尔文, 222 = ANSI泰语, 238 = ANSI拉丁语II (中欧), 255 = OEM Latin I

colour_index=

“[调色板](#)”中给出了“颜色指数”的解释; [颜色索引](#)。

escapement=

1 =上标, 2 =下标。

family=

价值观:

0 =无 (未知或不关心) 1 =罗马 (可变宽度, 无衬线) 2 =瑞士 (可变宽度, 无衬线) 3 =现代 (固定宽度, 带衬线或无衬线) 4 =脚本 (草书) 5 =装饰 (专业, 例如古英语, Fraktur)

font_index=

基于0的索引用于引用此Font () 实例。注意, 从不使用索引4; xlrld提供一个假的占位符。

height=

字体高度 (以缇为单位)。一个twip = 1/20的一个点。

italic=

1 =字符是斜体。

name=

字体的名称。示例: "Arial"。

struck_out=

1 =字符被删除。

underline_type=

价值观:

0 =无1 =单; 0x21 (33) =单一会计2 =双倍; 0x22 (34) =双重会计

underlined=

1 =字符加下划线。多余的; 看 underline_type 属性。

weight=

字体粗细 (100-1000)。普通文本的标准值为400, 粗体文本的标准值为700。

outline=

1 =字体是大纲样式（仅限Macintosh）

shadow=

1 =字体是阴影样式（仅限Macintosh）

class `xlrd.formatting.Format` (*format_key, ty, format_str*)

记录中的“数字格式”信息 `FORMAT`。

版本0.6.1中的新功能。

format_key=

关键是进入 `format_map`

type=

从格式字符串推断出的分类。目前，这仅用于区分数字和日期。价值观：

```
FUN = 0 # unknown
FDT = 1 # date
FNU = 2 # number
FGE = 3 # general
FTX = 4 # text
```

format_str=

格式字符串

`xlrd.formatting.fmt_bracketed_sub` ()

返回通过替换`repl`替换字符串中最左边的非重叠模式而获得的字符串。

类 `xlrd.formatting.XFBorder`

`XF` 记录的边界相关属性的集合。项目对应于Excel UI的格式 -> 单元格 -> 边框选项卡中的项目。

“[调色板](#)”中给出了“颜色指数”的解释; [颜色索引](#)。

有五种线型属性; 可能的值和相关含义是：

```
0 = No line,  
1 = Thin,  
2 = Medium,  
3 = Dashed,  
4 = Dotted,  
5 = Thick,  
6 = Double,  
7 = Hair,  
8 = Medium dashed,  
9 = Thin dash-dotted,  
10 = Medium dash-dotted,  
11 = Thin dash-dot-dotted,  
12 = Medium dash-dot-dotted,  
13 = Slanted medium dash-dotted.
```

换行样式8到13仅出现在BIFF8文件（Excel 97及更高版本）中。有关线型的图片，请参阅OOo docs s3.10（第22页）“单元格边框的线型（BIFF3-BIFF8）”。</p>

版本0.6.1 中的新功能。

top_colour_index=

单元格顶行的颜色索引

bottom_colour_index=

单元格底线的颜色索引

left_colour_index=

单元格左侧行的颜色索引

right_colour_index=

单元格右侧的颜色索引

diag_colour_index=

单元格对角线的颜色索引（如果有）

top_line_style=

单元格顶线的线条样式

bottom_line_style=

单元格底线的线条样式

left_line_style=

单元格左侧行的线条样式

right_line_style=

单元格右线的线条样式

diag_line_style=

单元格对角线的线条样式（如果有）

diag_down=

1 =从左上角到右下角绘制一条对角线

diag_up=

1 =从左下角到右上角画一条对角线

类 xlrd.formatting.XFBackground

XF 记录的背景相关属性的集合。项目对应于Excel UI的格式 - >单元格 - >模式选项卡中的项目。

“[调色板](#)”中给出了“颜色指数”的解释; [颜色索引](#)。

版本0.6.1 中的新功能。

fill_pattern=

请参阅OOo文档的第3.11节。

background_colour_index=

请参阅OOo文档的第3.11节。

pattern_colour_index=

请参阅OOo文档的第3.11节。

类 xlrd.formatting.XFAlignment

XF 记录的对齐和类似属性的集合。项目对应于Excel UI的格式 - >单元格 - >对齐选项卡中的项目。

版本0.6.1 中的新功能。

hor_align=

值：OOo文档的第6.115节（第214页）

vert_align=

值：OOo文档的第6.115节（第215页）

rotation=

值：OOo文档的第6.115节（第215页）。

❗ 注意

文件版本BIFF7和更早版本使用文档 `orientation` 属性; 这将被映射（无损失） `rotation` 。

text_wrapped=

1 =文本被包裹在右边距

indent_level=

一个数字 `range(15)`。

shrink_to_fit=

1 =缩小字体大小以使文本适合单元格。

text_direction=

0 =根据上下文; 1 =从左到右; 2 =从右到左

类 `xlrd.formatting.XFProtection`

XF 记录的保护相关属性的集合。项目对应于Excel UI格式 - >单元格 - >保护选项卡中的项目。请注意，OOo文档包含此捆绑属性中的“单元格或样式”位。这是不正确的; 该位用于确定要使用的捆绑包。

版本0.6.1中的新功能。

cell_locked=

1 =防止更改，移动，调整大小或删除单元格（仅当工作表受到保护时）。

formula_hidden=

1 =隐藏公式，以便在选择单元格时它不会出现在公式栏中（仅当工作表受到保护时）。

类 `xlrd.formatting.XF`

单元格，行，列和样式的扩展格式信息。

下面6个标志中的每一个都描述了特定属性组的有效性。

在单元格XF中：

- `flag==0` 表示 **XF** 使用父样式的属性，（但仅当属性在那里有效时）；
- `flag==1` 表示使用它的属性 **XF**。

风格XF：

- `flag==0` 表示属性设置有效;
- `flag==1` 表示应忽略该属性。

❗ 注意

API提供“原始”XF和“计算”XF。在后一种情况下，单元格XF已应用上述继承机制。

版本0.6.1中的新功能。

is_style=

0 =单元格XF, 1 =样式XF

parent_style_index=

cell XF: 索引到这个XF样式XF的Book.xf_list

样式XF: 0xFFFF

xf_index=

索引到 `xf_list`

font_index=

索引到 `font_list`

format_key=

关键是 `format_map`

警告

XF记录上的OOo文档称之为“索引格式记录”。它不是Python意义上的索引。它是地图的关键。这是真的只Excel 4.0和更早版本的文件，关键为format_map从XF实例是一样的索引format_list，并且只有当指数小于164。

protection=**无**

`XFProtection` 对象的实例。

background=**无**

`XFBackground` 对象的实例。

alignment=**无**

`XFAAlignment` 对象的实例。

border=**无**

`XFBorder` 对象的实例。

xlrd.formula

用于解析/评估Microsoft Excel公式的模块。

类 `xlrd.formula.Operand` (*akind = 无, 安勤= 无, arank = 0, atext = '?'*)

用于评估公式。下表描述了各种类型及其值的表示方式。

善良的 象征	种类 号码	价值表示
-----------	----------	------

oBOOL	3	整数：0 =>假; 1 =>真
OERR	4	无，或int错误代码（与Cell类中的XL_CELL_ERROR相同）。
oMSNG	五	由Excel用作缺少（未提供）函数参数的占位符。应该*不*作为最终公式结果出现。值为None。
oNUM	2	一个漂浮物。请注意，无法区分日期。
OREF	-1	该值为None或绝对Ref3D实例的非空列表。
奥廖尔	-2	值为None或完全或部分相对Ref3D实例的非空列表。
oSTRG	1	Unicode字符串。
oUNK	0	那种不明或含糊不清。值为None

kind=

oUNK意味着那种操作数是不明确的。

value=无

None表示操作数的实际值是变量（取决于单元格数据），而不是常量。

text='?'

原始公式的重组文本。函数名称将使用英语，而不考虑原始语言，这似乎不会记录在任何地方。分隔符是“，”，而不是“;”或其他任何可能更适合最终用户的语言环境分隔符; 补丁欢迎。

class xlrd.formula.Ref3D (tuple)

表示对一个或多个单元格的框的绝对或相对三维引用。

该 **coords** 属性是以下形式的元组：

```
(shtxlo, shtxhi, rowxlo, rowxhi, colxlo, colxhi)
```

在那里。 `0 <= thingxlo <= thingx < thingxhi`

！ 注意

很有可能; 例如， 可以具有和/或 不管工作表中实际使用了多少列/行。调用者需要决定如何处理这种情况。关键字: `thingx > nthings` `Print_Titles` `colxhi == 256` `rowxhi == 65536` `IndexError`

在COORDS属性的组件也可作为单独的属性: `shtxlo` , `shtxhi` , `rowxlo` , `rowxhi` , `colxlo` , 和 `colxhi` 。

该 **relflags** 属性是6元组标志，表示相应的（列）（lo | hi）是相对（1）还是绝对（0）。

❗ 注意

没有关于参考可能相对于哪个细胞的信息。调用者必须决定 `oREL` 操作数的用途。

版本0.6.0中的新功能。

`xlrd.formula.cellname (一行x, COLX)`

效用函数: `=>` `(5, 7)` `'H6'`

`xlrd.formula.cellnameabs (一行x, COLX, R1C1 = 0)`

效用函数: `=>` `(5, 7)` `'H6'`

`xlrd.formula.colname (colx)`

效用函数: `7` `=>` `'H'`, `27` `=>` `'AB'`

`xlrd.formula.rangename3d (书, ref3d)`

效用函数: `=>` (假设Excel的默认工作表名

称) `Ref3D(1, 4, 5, 20, 7, 10)` `'Sheet2:Sheet3!H6:J20'`

`xlrd.formula.rangename3drel (book, ref3d, browx = None, bcolx = None, r1c1 = 0)`

实用功能: `Ref3D(coords=(0, 1, -32, -22, -13, 13), relflags=(0, 0, 1, 1, 1, 1))`

在R1C1模式下`=>` `'Sheet1!R[-32]C[-13]:R[-23]C[12]'`

在A1模式下, `=>`取决于基本单元格 `(browx, bcolx)`

xlrd.sheet

班级 `xlrd.sheet.Sheet (书, 职位, 姓名, 号码)`

包含一个工作表的数据。

在单元格访问函数中, `rowx` 是一个从零开始计数的行索引, `colx` 是一个从零开始计数的列索引。以预期方式支持行/列索引和切片位置的负值。

有关单元格类型和单元格值的信息, 请参阅 `Cell` 该类的文档。

❗ 警告

您不自己实例化此类。您可以 `Sheet` 通过 `Book` 调用时返回的对象访问对象 `xlrd.open_workbook()`。

col (`colx`)

返回 `Cell` 给定列中的对象序列。

gcw

一个256元素的元组，对应于此工作表的GCW记录的内容。如果没有这样的记录，则视为所有位为零。仅适用于BIFF4-7。请参阅 `Colinfo` 课程文档以供讨论。

vert_split_pos=

左窗格中的列数（冻结窗格;对于拆分窗格，请参阅代码中的注释）

horz_split_pos=

顶部窗格中的行数（冻结窗格;对于拆分窗格，请参阅代码中的注释）

horz_split_first_visible=

底部冻结/拆分窗格中第一个可见行的索引

vert_split_first_visible=

右侧冻结/拆分窗格中第一个可见列的索引

split_active_pane=

冷冻窗格：忽略它。拆分窗格：OOo文档中的说明和图表。

has_pane_record=

指定 `PANE` 记录是否存在的布尔值，忽略除非你是 `xlutils.copy`

book=~~无~~

`Book` 对此工作表所属对象的引用。

用法示例： `some_sheet.book.datemode`

name=""

表单名称。

nrows=

工作表中的行数。行索引在 `range(thesheet.nrows)` 。

ncols=

工作表中的标称列数。它比找到的最大列索引多一个，忽略尾随空单元格。另请参阅 和 的 `ragged_rows` 参数。 `open_workbook()` `row_len()`

defcolwidth=~~无~~

`DEFCOLWIDTH` 记录的默认列宽，否则 `None`。来自OOo文档：

列宽为字符，使用默认字体的零字符宽度（文件中的第一个FONT记录）。Excel会为默认宽度添加一些额外空间，具体取决于默认字体和默认字体大小。如何精确计算得到的列宽的算法尚不清楚。示例：使用Arial字体（大小为10磅），此记录中的默认宽度8设置导致列宽为8.43。

对于默认层次结构，请参阅 `Colinfo` 类。

版本0.6.1中的新功能。

`standardwidth=无`

`STANDARDWIDTH` 记录的默认列宽，否则 `None`。

来自OOo文档：

默认宽度为零字符宽度的1/256，使用默认字体（文件中的第一个FONT记录）。

对于默认层次结构，请参阅 `Colinfo` 类。

版本0.6.1中的新功能。

`default_row_height=无`

如果 `ROW` 该行没有记录，则用于行的默认值。从 *可选* `DEFAULTROWHEIGHT` 记录中。

`default_row_height_mismatch=无`

如果 `ROW` 该行没有记录，则用于行的默认值。从 *可选* `DEFAULTROWHEIGHT` 记录中。

`default_row_hidden=无`

如果 `ROW` 该行没有记录，则用于行的默认值。从 *可选* `DEFAULTROWHEIGHT` 记录中。

`default_additional_space_above=无`

如果 `ROW` 该行没有记录，则用于行的默认值。从 *可选* `DEFAULTROWHEIGHT` 记录中。

`default_additional_space_below=无`

如果 `ROW` 该行没有记录，则用于行的默认值。从 *可选* `DEFAULTROWHEIGHT` 记录中。

`colinfo_map= {}`

从列索引到 `Colinfo` 对象的映射。通常 `COLINFO`，所有列索引的记录中都有一个条目 `range(257)`。

❗ 注意

xlrd忽略不存在的第257列的条目。

另一方面，未使用的列可能没有条目。

版本0.6.1中的新功能。

仅在填充时填充 `open_workbook(..., formatting_info=True)`

`rowinfo_map= {}`

从行索引到 `Rowinfo` 对象的映射。

..注意: :

可能缺少条目 - 至少有一个XLS文件源不会打扰写入 `ROW` 记录。

版本0.6.1中的新功能。

仅在填充时填充 `open_workbook(..., formatting_info=True)`

`col_label_ranges= []`

包含列标签的单元格的地址范围列表。这些是通过“插入”>“名称”>“标签”>“列”在Excel中设置的。

版本0.6.0中的新功能。

如何解构列表:

```
for crange in thesheet.col_label_ranges:
    rlo, rhi, clo, chi = crange
    for rx in xrange(rlo, rhi):
        for cx in xrange(clo, chi):
            print "Column label at (rowx=%d, colx=%d) is %r" \
                  (rx, cx, thesheet.cell_value(rx, cx))
```

`row_label_ranges= []`

包含行标签的单元格的地址范围列表。有关详细信息, 请参阅 `col_label_ranges`。

版本0.6.0中的新功能。

`merged_cells= []`

已合并的单元格的地址范围列表。这些是通过格式>单元格>对齐在Excel中设置, 然后勾选“合并单元格”框。

❗ 注意

上限是独占的: 即仅跨越两个单元格。 `[2, 3, 7, 9]`

❗ 注意

仅在提取时提取 `open_workbook(..., formatting_info=True)`

版本0.6.1中的新功能。

如何解构列表:

```

for crange in thesheet.merged_cells:
    rlo, rhi, clo, chi = crange
    for rowx in xrange(rlo, rhi):
        for colx in xrange(clo, chi):
            # cell (rlo, clo) (the top left one) will carry the data
            # and formatting info; the remainder will be recorded as
            # blank cells, but a renderer will apply the formatting info
            # for the top left cell (e.g. border, pattern) to all cells in
            # the range.

```

rich_text_runlist_map= {}

映射到元组列表。偏移量定义字符串开始使用的位置。预计抵消将按升序排列。如果第一个偏移量不为零，则意味着应该从偏移量0使用单元格的字体。`(rowx, colx)` `(offset, font_index)` `XF`

这是稀疏映射。没有使用富文本格式化的单元格没有条目。

如何使用：

```

runlist = thesheet.rich_text_runlist_map.get((rowx, colx))
if runlist:
    for offset, font_index in runlist:
        # do work here.
    pass

```

版本0.7.2中的新功能。

仅在填充时填充 `open_workbook(..., formatting_info=True)`

horizontal_page_breaks= []

此工作表中的水平分页符列表。中断是表单中的元组

。 `(index of row after break, start col index, end col index)`

仅在填充时填充 `open_workbook(..., formatting_info=True)`

版本0.7.2中的新功能。

vertical_page_breaks= []

此工作表中的垂直分页符列表。中断是表单中的元组

。 `(index of col after break, start row index, end row index)`

仅在填充时填充 `open_workbook(..., formatting_info=True)`

版本0.7.2中的新功能。

visibility=

表格的可见性：

0 =可见 1 =隐藏（可以被用户取消隐藏 - 格式 - >工作表 - >取消隐藏） 2 =“非常隐藏”（只能由VBA宏取消隐藏）。

hyperlink_list= []

与工作表中找到的记录 `Hyperlink` 对应的对象列表 `HLINK`。

版本0.7.2 中的新功能。

hyperlink_map= {}

稀疏映射到中的项目。未映射超链接未覆盖的单元格。可以使用Excel UI设置覆盖大于1x1矩形单元格的超链接。超链接矩形可能重叠（Excel不检查）。单击一个多重覆盖的单元格时，激活的超链接（以及此处映射的超链接）是最后一个

。 `(rowx, colx)` `hyperlink_list` `hyperlink_list`

版本0.7.2 中的新功能。

cell_note_map= {}

从对象到对象的稀疏映射。不映射不包含注释（“注释”）的单元格。 `(rowx, colx)` `Note`

版本0.7.2 中的新功能。

cell (一行x, COLX)

`Cell` 给定行和列中的对象。

cell_value (一行x, COLX)

给定行和列中单元格的值。

cell_type (一行x, COLX)

给定行和列中单元格的类型。

请参阅 `Cell` 课程文档。

cell_xf_index (一行x, COLX)

给定行和列中单元格的XF索引。这是一个索引 `xf_list`。

版本0.6.1 中的新功能。

row_len (rowx)

返回给定行中的有效单元格数。对于使用 `open_workbook(ragged_rows=True)` 它可能会产生少于 `ncols` 单元格的行。

版本0.7.2 中的新功能。

row (rowx)

返回 `Cell` 给定行中的对象序列。

get_rows ()

返回一个生成器，用于遍历每一行。

row_types (*一行*, *start_colx = 0*, *end_colx = 无*)

返回给定行中单元格类型的切片。

row_values (*一行*, *start_colx = 0*, *end_colx = 无*)

返回给定行中单元格值的切片。

row_slice (*一行*, *start_colx = 0*, *end_colx = 无*)

返回 `Cell` 给定行中对象的切片。

col_slice (*colx*, *start_rowx = 0*, *end_rowx = 无*)

返回 `Cell` 给定列中对象的切片。

col_values (*colx*, *start_rowx = 0*, *end_rowx = 无*)

返回给定列中单元格值的切片。

col_types (*colx*, *start_rowx = 0*, *end_rowx = 无*)

返回给定列中单元格类型的切片。

computed_column_width (*colx*)

确定列显示宽度。

参数: *colx* - 查询列的索引, 范围为0到255.请注意, 可以找出用于显示没有单元格信息的列的宽度, 例如列IV (*colx* = 255) 。

返回: 用于按Excel显示给定列的列宽, 以标准字符宽度的1/256 (第一种字体中的数字零) 为单位。

版本0.6.1中的新功能。

类 `xlrd.sheet.MSODrawing`

类 `xlrd.sheet.MSObj`

类 `xlrd.sheet.MSTxo`

类 `xlrd.sheet.Note`

表示用户“注释”或“注释”。注意对象可通过 `Sheet.cell_note_map` 。

版本0.7.2中的新功能。

author=""

作者注意

col_hidden=

True 如果包含列被隐藏

colx=

列索引

rich_text_runlist=无

元组列表。与之不同，第一个偏移量应始终为

0。 `(offset_in_string, font_index)` `Sheet.rich_text_runlist_map`

row_hidden=

如果隐藏包含行，则为True

rowx=

行索引

show=

如果始终显示注释，则为True

text=

笔记的文字

类 `xlrd.sheet.Hyperlink`

包含超链接的属性。可以通过 `Sheet.hyperlink_list` 和访问超链接对象 `Sheet.hyperlink_map`。

版本0.7.2中的新功能。

frowx=无

第一行索引

lrowx=无

最后一行的索引

fcplx=无

第一列索引

lcolx=无

最后一栏的索引

type=无

超链接类型。Unicode字符串，'url'，'unc'，'本地文件'，'工作簿'，'未知'之一

url_or_path=无

URL或文件路径，具体取决于类型。Unicode字符串，除了少见的本地但不存在的名称中包含非ASCII字符的文件，在这种情况下只有“8.3”文件名可用，如 `bytes`（3.x）或 `str`（2.x）字符串，*编码未知*。

desc=无

描述。这将显示在单元格中，并且应与单元格值相同。Unicode字符串，或 `None`。似乎不可能没有Excel UI创建的描述。

target=无

目标框架。Unicode字符串。

❗ 注意

在野外没有发现过这种情况。在Excel UI中创建一个似乎是不可能的。

textmark=无

“http://docs.python.org/library#struct_module”中“#”后面的 `Sheet1!A1:Z99` 部分，或者类型为“工作簿”时的部分。

quicktip=无

当光标悬停在超链接上时显示“快速提示”的文本。

class `xlrd.sheet.Cell` (`ctype`, `value`, `xf_index` = `None`)

包含一个单元格的数据。

❗ 警告

你不要自己打这个班。您可以 `Cell` 通过在调用时返回 `Sheet` 的 `Book` 对象中找到的 对象的方法 来访问对象 `open_workbook()`

`Cell`对象有三个属性：`ctype` 是int，`value`（取决于 `ctype`）和 `xf_index`。如果 `formatting_info` 在打开工作簿时未启用，`xf_index` 则将 `None`。

下表描述了单元格的类型以及它们的值在Python中的表示方式。

输入符号	型号	Python值
XL_CELL_EMPTY	0	空字符串"
XL_CELL_TEXT	1	Unicode字符串
XL_CELL_NUMBER	2	浮动
XL_CELL_DATE	3	浮动

XL_CELL_BOOLEAN	4	INT; 1表示TRUE, 0表示FALSE
XL_CELL_ERROR	五	int表示内部Excel代码; 有关文本表示, 请参阅提供的字典 error_text_from_code
XL_CELL_BLANK	6	空字符串"。注意: 仅当使用open_workbook (... , formatting_info = True) 时才会出现此类型。

类 xlrd.sheet.Colinfo

适用于工作表中一个或多个列的宽度和默认格式设置信息。源于 COLINFO 记录。

根据OOo文档, 这是宽度的默认层次结构:

在BIFF3中, 如果 COLINFO 列缺少记录, 则使用记录中指定的宽度 DEFCOLWIDTH 。

在BIFF4-BIFF7中, COLINFO 仅在 GCW 记录中清除此列的相应位时才使用此记录中设置的宽度, 否则将 DEFCOLWIDTH 使用记录中设置的列宽 (STANDARDWIDTH 在这种情况下, 记录始终被忽略^[1]) 。

在BIFF8中, 如果 COLINFO 列缺少记录, 则使用记录中指定的宽度 STANDARDWIDTH 。

如果 STANDARDWIDTH 还缺少此记录, DEFCOLWIDTH 则使用记录的列宽。

[1] GCW 记录中的文档说:

如果设置了一个位, 则相应的列使用 STANDARDWIDTH 记录中设置的宽度。如果清除了某个位, 则相应的列将使用 COLINFO 此列记录中设置的宽度。

如果设置了一个位, 并且工作表不包含该 STANDARDWIDTH 记录, 或者该位被清除, 并且该工作表不包含该 COLINFO 记录, DEFCOLWIDTH 则将使用该工作表的记录。

xlrd与故事的GCW版本一致。对源的引用可能很有用: 请参阅 Sheet.computed_column_width() 。

版本0.6.1 中的新功能。

width=

列的宽度, 以零字符宽度的1/256为单位, 使用默认字体 (FONT 文件中的第一个记录) 。

xf_index=

用于格式化空单元格的XF索引。

hidden=

1 =隐藏列

bit1_flag=

1位标志的值, 其目的是未知但通常被视为设置为1

outline_level=

列的大纲级别, 在 range(7) 。

(0 =没有大纲)

`collapsed=`

1 =列已折叠

类 `xlrd.sheet.Rowinfo`

应用于工作表中某行的高度和默认格式设置信息。源于 `ROW` 记录。

版本0.6.1 中的新功能。

`height`

排的高度，以缇为单位。一个twip == 1/20的一个点。

`has_default_height`

0 =行有自定义高度; 1 =行具有默认高度。

`outline_level`

行的大纲级别 (0到7)

`outline_group_starts_ends`

1 =大纲组在此处开始或结束 (取决于大纲按钮所在的位置, 请参阅 `WSBOOL` 记录, xlrd未解析), 并且已折叠。

`hidden`

1 =隐藏行 (手动, 或通过过滤器或大纲组)

`height_mismatch`

1 =行高和默认字体高度不匹配。

`has_default_xf_index`

1 = xf_index属性可用; 0 =忽略它。

`xf_index`

`XF` 此行中空单元格的默认记录索引。如果, 请不要使用此。 `has_default_xf_index == 0`

`additional_space_above`

如果此行中至少一个单元格的上边框或上面一行中至少一个单元格的下边框使用粗线样式格式化, 则设置此标志。不考虑细线和中线样式。

`additional_space_below`

如果此行中至少一个单元格的下边框或下面行中至少一个单元格的上边框使用中等或粗线样式格式化, 则设置此标志。不考虑细线样式。

xlrd.xldate

用于在Excel文件中处理日期和时间的工具。

转换 `days` 为从一个完整的“julian day number”（即JDN）开始。FWIW: `(year, month, day)`

- JDN 0对应于11月24日星期一的格里高利年-4713。

更重要的是：

- 格里高利1900-03-01中的中午（基于1900的系统中的第61天）是JDN 2415080.0
- 格里高利1904-01-02中的中午（基于1904年的系统中的第1天）是JDN 2416482.0

异常 `xlrd.xldate.XLDateError`

所有与日期时间相关的错误的基类。

异常 `xlrd.xldate.XLDateNegative`

`xldate < 0.00`

异常 `xlrd.xldate.XLDateAmbiguous`

1900年的闰年问题 `(datemode == 0 and 1.0 <= xldate < 61.0)`

异常 `xlrd.xldate.XLDateTooLarge`

格里高利年10000或以后

异常 `xlrd.xldate.XLDateBadDatemode`

`datemode` arg既不是0也不是1

异常 `xlrd.xldate.XLDateBadTuple`

`xlrd.xldate.xldate_as_tuple (xldate, datemode)`

将Excel编号（假定为代表日期，日期时间或时间）转换为适合于提供给datetime或mx.DateTime构造函数的元组。

- 参数：
- `xldate` - Excel编号
 - `datemode` - 基于0: 1900, 基于1: 1904。

- 举：
- `xlrd.xldate.XLDateNegative` -
 - `xlrd.xldate.XLDateAmbiguous` -
 - `xlrd.xldate.XLDateTooLarge` -
 - `xlrd.xldate.XLDateBadDatemode` -
 - `xlrd.xldate.XLDateError` -

返回： 格里高利。 `(year, month, day, hour, minute, nearest_second)`

⚠ 警告

使用此函数解释工作簿的内容时，应传入 `datemode` 该工作簿的属性。工作簿是否曾经在 Macintosh 附近无关紧要。

❗ 特殊情况

如果，则假定代表时间; 将被退回。 `0.0 <= xldate < 1.0 (0, 0, 0, hour, minute, second)`

⚠ 注意

`1904-01-01` 不被视为系统中的有效日期 `datemode==1` ; 它的“序列号”为零。

`xlrd.xldate.xldate_as_datetime (xldate, datemode)`

将 Excel 日期/时间编号转换为 `datetime.datetime` 对象。

参数:

- `xldate` - Excel 编号
- `datemode` - 基于 0: 1900, 基于 1: 1904。

返回: 一个 `datetime.datetime` 对象。

`xlrd.xldate.xldate_from_date_tuple (date_tuple, datemode)`

将日期元组 (年, 月, 日) 转换为 Excel 日期。

参数:

- **年** - 格里高利年。
- **月** - `1 <= month <= 12`
- **一天** - `1 <= day <= last day of that (year, month)`
- `datemode` - 基于 0: 1900, 基于 1: 1904。

举:

- `xlrd.xldate.XLDateAmbiguous` -
- `xlrd.xldate.XLDateBadDatemode` -
- `xlrd.xldate.XLDateBadTuple` - 太早/晚或组件无效 `(year, month, day)`
- `xlrd.xldate.XLDateError` -

`xlrd.xldate.xldate_from_time_tuple (time_tuple)`

将时间元组转换为 Excel “日期”值 (一天的分数) 。 `(hour, minute, second)`

参数:

- **小时** - `0 <= hour < 24`
- **分钟** - `0 <= minute < 60`
- **第二** - `0 <= second < 60`

举: `xlrd.xldate.XLDateBadTuple` - 超出范围的小时, 分钟或秒

`xlrd.xldate.xldate_from_datetime_tuple` (*datetime_tuple*, *datemode*)

将日期时间元组转换为Excel日期值。有关更多详细信息，请参阅其他*xldate_from _ * _*元组函数。 (*year*, *month*, *day*, *hour*, *minute*, *second*)

参数:

- *datetime_tuple* - (*year*, *month*, *day*, *hour*, *minute*, *second*)
- *datemode* - 基于0: 1900, 基于1: 1904。