

概述

本文基于openpyxl封装一个excel解析类，请注意，不采用Python的任何高级特性，就简简单单的一个类，实现excel的一些基本操作，并演示如何使用该类。

封装类及使用

直接上代码。

```
#-*- coding:utf-8 -*-

__author__ = "苦叶子"

import os
from openpyxl import Workbook, load_workbook

# excel自定义封装类
class LYMOpenXL:
    def __init__(self, path, read_only=False):
        self.wb = None
        if os.path.exists(path):
            self.path = path
            self.wb = load_workbook(self.path, read_only=read_only)
        else:
            print("%s 文件不存在" % path)
            exit(0)

    # 获取excel的行数
    # 如果指定的工作簿存在，则返回其数据行数，否则返回None
    def get_cell_row(self, sheet):
        if self.wb:
            # 先通过sheet获取工作簿
            sh = self.wb.get_sheet_by_name(sheet)
            if sh:
                return sh.max_row

        return None

    # 获取excel的列数
    # 如果指定的工作簿存在，则返回其数据列数，否则返回None
    def get_cell_col(self, sheet):
        if self.wb:
            # 先通过sheet获取工作簿
            sh = self.wb.get_sheet_by_name(sheet)
            if sh:
                return sh.max_column

        return None
```

```

# 获取工作簿名称列表
def get_sheets_name(self):
    if self.wb:
        return self.wb.get_sheet_names()

    return None

# 通过索引获取工作簿名
# 索引从0开始
def get_sheet_name_by_index(self, index):
    if self.wb:
        sheets = self.wb.get_sheet_names()
        sheet_len = len(sheets)
        if index >= 0 and index < sheet_len:
            return sheets[index]

    return None

# 创建工作簿
def create_sheet(self, name, index=0):
    res = False
    if self.wb:
        self.wb.create_sheet(title=name, index=index)
        res = True

    return res

# 修改工作簿名
def set_sheet_name(self, sheet_name, name):
    res = False
    if self.wb:
        self.wb[sheet_name].title = name
        res = True

    return res

# 获取单元格值
def get_cell_value(self, sheet, row, col):
    value = None
    if self.wb:
        value = self.wb[sheet].cell(row=row, column=col).value

    return value

# 设置单元格值
def set_cell_value(self, sheet, row, col, value):
    res = False
    if self.wb:
        self.wb[sheet].cell(row=row, column=col).value = value
        res = True

```

```

        return res

# 保存
def save(self, path=""):
    if path != "":
        self.path = path

    if self.wb:
        self.wb.save(self.path)

if __name__ == "__main__":
    print("python openpyxl基本实例")
    print("---" * 20, end='\n')

    xl = LYMOpenXL("openpyxl_demo.xlsx")

    # 获取所有工作簿名
    sheets = xl.get_sheets_name()
    print(">>>获取工作簿列表", end='\n')
    print(sheets)

    # 通过索引获取工作簿名
    print("---" * 20, end='\n')
    print(">>>通过所有获取工作簿名")
    for index in range(0, len(sheets)):
        print(xl.get_sheet_name_by_index(index), end=' ')
    print(end='\n')

    # 获取各工作簿数据行列数
    print("---" * 20)
    for sheet in sheets:
        nrows = xl.get_cell_row(sheet)
        ncols = xl.get_cell_col(sheet)
        print("工作簿[%s]的数据行列数为(%d, %d)" % (sheet, nrows, ncols))

    # 获取各工作簿中的数据
    print("---" * 20)
    print(">>>获取工作簿中数据")
    for sheet in sheets:
        nrows = xl.get_cell_row(sheet)
        ncols = xl.get_cell_col(sheet)
        print("---" * 20, end='\n')
        print("工作簿[%s]数据如下: " % sheet)
        for row in range(1, nrows+1):
            for col in range(1, ncols+1):
                value = xl.get_cell_value(sheet, row, col)
                print("[%d, %d]->%s" % (row, col, value), end='\t')

    # 修改各工作簿第一行的数据为: DeepTest
    print("---" * 20)
    print(">>>设置工作簿中数据")
    for sheet in sheets:

```

```
nrows = xl.get_cell_row(sheet)
ncols = xl.get_cell_col(sheet)
for col in range(1, ncols+1):
    xl.set_cell_value(sheet, row=1, col=col, value="DeepTest")

# 保存
xl.save()
```

小结

请自行查阅openpyxl官方文档去实现以下功能

- 一次性读取多个单元格的值
- 设置指定单元格的字体
- 设置指定单元格的样式（例如背景色等）

扫一扫关注微信公众号：

