# Openpyxl 操作EXCEL

## 打开excel wb= load\_workbook(filename, read\_only=False, keep\_vba=KEEP\_VBA, data\_only=False, keep\_links=True)

返回rtype : class: openpyxl.workbook.Workbook

例子:wb=openpyxl.load\_workbook(r‘D:\test.txt’)

## 取得excel所有sheet名字 wb.get\_sheet\_names()

返回：一个包含所有sheet名字的List

例子：for sheetName in wb.get\_sheet\_names():

## 返回某个sheet wb[sheetName]

返回：一个sheet对象

例子：ws=wb[‘sheet1’]

## 删除某个sheet wb.remove（sheet对象）

例子：wb.remove(wb[‘sheet1’])

## 获取一个sheet的最大行 sheet.max\_row

## 获取sheet中的一个cell sheet.**cell**(self, row, column, value=None)或者sheet[‘D5’]

返回：一个cell对象

例子：sheet.cell(row=i+1,column=13).value='haha'

或者：sheet[‘D5’]

## 取得或者设置一个单元格的值 cell.**value**

例子：sheet.cell(row=i+1,column=13).value='haha'

## 修改完成后保存到文件 wb.save(filename)

## 合并格单元说明

对于合并格子单元，读取的时候，只有左上角那个会有值，其它的都是None

# Os 操作文件和目录

## 取得一个目录下所有文件和目录名字 os.listdir(rootPath)

## 判断是否是目录 os.path.isdir(filename)

# 文件读写

## 读取一个文件 **open(filename)**

返回：一个file对象

**例子：helloFile = open('C:\\Users\\*your\_home\_folder*\\hello.txt')**

# 日期和时间

## 日期时间 datetime

包含**datetime类，date类，time类.**

### 比较大小

\_\_eq\_\_(self, value,)

返回：self==value.

**\_**\_ge\_\_(self, value)

返回：self>=value.

\_gt\_\_(self, value, /)

返回：self>value.

### 两个datetime相减

返回：timedelta对象，该对象包含：

days

Number of days.

microseconds

Number of microseconds (>= 0 and less than 1 second).

seconds

Number of seconds (>= 0 and less than 1 day).

### 间隔时间转化为秒 timedelta.total\_seconds(...)

返回：秒。可以为负数。

## 方便的时间格式化工具 time

每个时间戳都以自从1970年1月1日午夜（历元）经过了多长时间来表示。

Python 的 time 模块下有很多函数可以转换常见日期格式。如函数time.time()用于获取当前时间戳,

### time.time()

返回：当前时间的时间戳(秒)

例子ticks = time.time()

print "当前时间戳为:", ticks

返回：当前时间戳为: 1459994552.51

### time.localtime(second)

参数second: 时间戳，秒。不带参数表示当前时间

返回：一个struct\_time元组

|  |  |  |
| --- | --- | --- |
| **序号** | **属性** | **值** |
| 0 | tm\_year | 2008 |
| 1 | tm\_mon | 1 到 12 |
| 2 | tm\_mday | 1 到 31 |
| 3 | tm\_hour | 0 到 23 |
| 4 | tm\_min | 0 到 59 |
| 5 | tm\_sec | 0 到 61 (60或61 是闰秒) |
| 6 | tm\_wday | 0到6 (0是周一) |
| 7 | tm\_yday | 1 到 366(儒略历) |
| 8 | tm\_isdst | -1, 0, 1, -1是决定是否为夏令时的旗帜 |

例子：

import time

localtime = time.localtime(time.time())

print "本地时间为 :", localtime

以上实例输出结果：

本地时间为 : time.struct\_time(tm\_year=2016, tm\_mon=4, tm\_mday=7, tm\_hour=10, tm\_min=3, tm\_sec=27, tm\_wday=3, tm\_yday=98, tm\_isdst=0)

### 格式化输出日期 time.strftime(format[, t])

格式化日期

例子：

# 格式化成2016-03-20 11:45:39形式

print time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())

# 格式化成Sat Mar 28 22:24:24 2016形式

print time.strftime("%a %b %d %H:%M:%S %Y", time.localtime())

# 将格式字符串转换为时间戳

a = "Sat Mar 28 22:24:24 2016" print time.mktime(time.strptime(a,"%a %b %d %H:%M:%S %Y"))

格式：

%Y  Year with century as a decimal number.  
%m  Month as a decimal number [01,12].  
%d  Day of the month as a decimal number [01,31].  
%H  Hour (24-hour clock) as a decimal number [00,23].  
%M  Minute as a decimal number [00,59].  
%S  Second as a decimal number [00,61].  
%z  Time zone offset from UTC.  
%a  Locale's abbreviated weekday name.  
%A  Locale's full weekday name.  
%b  Locale's abbreviated month name.  
%B  Locale's full month name.  
%c  Locale's appropriate date and time representation.  
%I  Hour (12-hour clock) as a decimal number [01,12].  
%p  Locale's equivalent of either AM or PM.

### 格式化输入日期 time. [strptime](http://localhost:12345/time.html#-strptime)(string, format)