# python入坑篇

**1.python xlrd 操作Excel读写**

- 打开Excel文件读取数据

data = xlrd.open\_workbook('excelFile.xls')

- 使用技巧

获取一个工作表

table = data.sheets()[0] #返回excel中所有的sheets的对象,list类型

table = data.sheet\_by\_index(0) #通过索引顺序获取

table = data.sheet\_by\_name(u'Sheet1') #通过名称获取

- 获取整行和整列的值（数组）

table.row\_values(i)

table.col\_values(i)

- 获取行数和列数

nrows = table.nrows

ncols = table.ncols

- 循环行列表数据

for i in range(nrows ):

print table.row\_values(i)

- 单元格

cell\_A1 = table.cell(0,0).value

cell\_C4 = table.cell(2,3).value

- 使用行列索引

cell\_A1 = table.row(0)[0].value

cell\_A2 = table.col(1)[0].value

简单的写入

row = 0

col = 0

# 类型 0 empty,1 string, 2 number, 3 date, 4 boolean, 5 error

ctype = 1 value = '单元格的值'

xf = 0 # 扩展的格式化

table.put\_cell(row, col, ctype, value, xf)

table.cell(0,0) #单元格的值'

table.cell(0,0).value #单元格的值'

openpyxl

wb.remove\_sheet(wb[‘new title’])

wb.worksheets #查看有多少sheet

wb.merge\_cells(“B2:G3”)

fill = PatternFill(“solid”, fgColor=”1784CD”)

ws[“D4”].fill = fill

font = Font(name=u”宋体”，size=14, color=RED, blod=True)

align = Alignment(horizontal=’center’, vertical=’center’)

ws.cellStyle(‘B2’, font, align)

ws.row\_dimensions[1].height=70

ws.column\_dimensions[‘B’].width = 20

for i in range(1, len(list(sheet.columns)[1])):

sheet.cell(row=i, column=1).value

1. **导出pip包**

导出：pip freeze > pip\_list.txt

安装通过文件：pip install -r pip\_list.txt

**3.操作数据库**

conn = pymysql.connect(

# host='192.168.1.102',

# user='root',

# password='root',

# # database='xxx',

# port=3308,

# charset='utf8'

# )

cursor = conn.cursor()

cursor.execute() //执行sql语句

conn.close()

connect对象支持的方法：

cursor()

commit()

rollback()

close()

***sqlalchemy***

info = db.session.execute(‘sql’)

info.fetchall() // 结果是list, 只能获取一次

for i in info:

print(i) // 结果是空

~~info.fetchall()~~

for i in info:

print(i) // 结果是想要的结果

**4.random 的使用**

import randpom

random.random() // 生成0-1的随机数

random.ranint(a,b) // 成成a-b的整数

lst = list(range(0, 100))

random.choice(lst) // 随机取样

.sample(lst, 5) // 随机取多个元素

.shuffle(lst) // 随机打乱

**5. time 的使用**

import time

for i in range(2)

print("hello")

time.sleep(1) // 1秒中输出一个‘hello’

print(time.time()) // 生成时间戳

print(time.ctime()) // 将当前时间转换成字符串，'str' > Sun Aug 26 08:40:05 2018

print(time.localtime()) // 生成时间结构类型

// -> time.struct\_time(tm\_year=2018, tm\_mon=8, tm\_mday=26, tm\_hour=8, tm\_min=34, tm\_sec=47,

tm\_wday=6, tm\_yday=238, tm\_isdst=0)

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

%y 两位数的年份表示（00-99）

%Y 四位数的年份表示（000-9999）

%m 月份（01-12）

%d 月内中的一天（0-31）

%H 24小时制小时数（0-23）

%I 12小时制小时数（01-12）

%M 分钟数（00=59）

%S 秒（00-59）

%a 本地简化星期名称

%A 本地完整星期名称

%b 本地简化的月份名称

%B 本地完整的月份名称

%c 本地相应的日期表示和时间表示

%j 年内的一天（001-366）

%p 本地A.M.或P.M.的等价符

%U 一年中的星期数（00-53）星期天为星期的开始

%w 星期（0-6），星期天为星期的开始

%W 一年中的星期数（00-53）星期一为星期的开始

%x 本地相应的日期表示

%X 本地相应的时间表示

%Z 当前时区的名称

%% %号本身

**6. 多线程，异步**

def async1(f):

def wrapper(\*args, \*\*kwargs):

thr = Thread(target=f, args=args, kwargs=kwargs)

thr.start()

return wrapper

@async1

def foo():

time.sleep(1)

print(\*\*\*\*)

# 编程技巧

