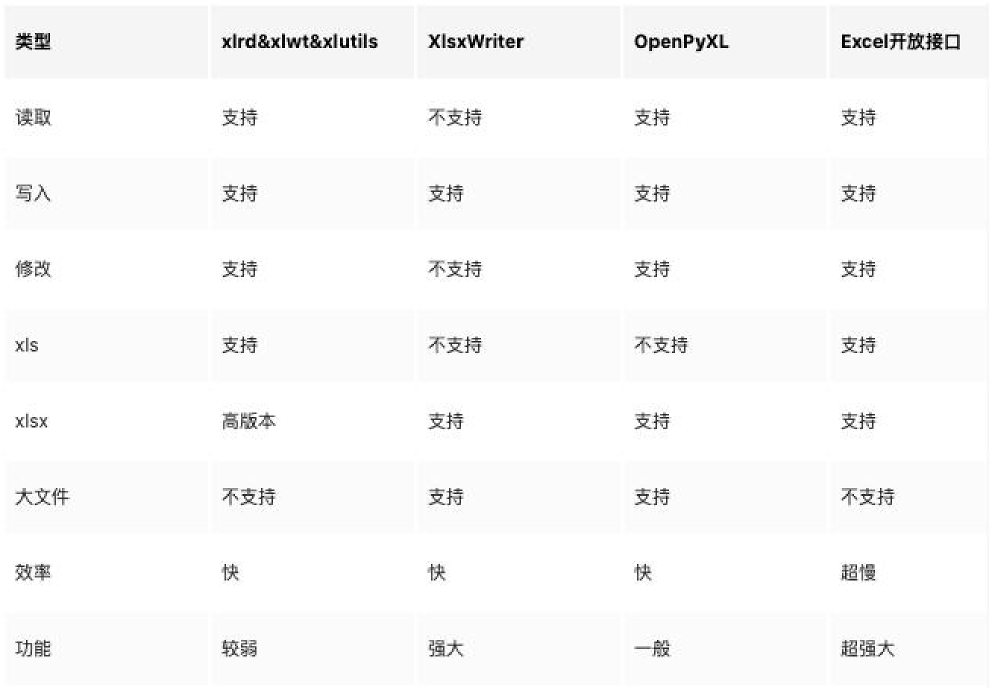
第一章：《从此做表不加班》Excel 自动化处理

office 家族其实都可以用 VBA 解决自动化的问题，但可能很多人不会用。

python 针对 excel 有很多的第三方库可以用，比如 xlwings、xlsxwriter、xlrd、xlwt、pandas、xlsxwriter、 win32com、xlutils 等等。

这些库可以很方便地实现对 excel 文件的增删改写、格式修改等，当然并不推荐你全部都去尝试一下，这样时间成本太大了。



类型	xlrd&xlwt&xlutils	XlsxWriter	OpenPyXL	Excel开放接口
读取	支持	不支持	支持	支持
写入	支持	支持	支持	支持
修改	支持	不支持	支持	支持
xls	支持	不支持	不支持	支持
xlsx	高版本	支持	支持	支持
大文件	不支持	支持	支持	不支持
效率	快	快	快	超慢
功能	较弱	强大	一般	超强大

xlrd：用于读取 Excel 文件；

xlwt：用于写入 Excel 文件；

xlutils：用于操作 Excel 文件的实用工具，比如复制、分割、筛选等

1.1 Excel 基本操作

写入数据

```
# pip install xlwt

# 导入模块
```

```
import xlwt
# 读入文件
wb = xlwt.Workbook()
# 增加工作薄
sh1 = wb.add_sheet('电影')
# 获取单元格
sh1.write(0, 0, '影片')
sh1.write(0, 1, '综合票房')
sh1.write(0, 2, '票房占比')
sh1.write(0, 3, '排片场次')

sh1.write(1, 0, '如果声音记不得')
sh1.write(1, 1, 361.57)
sh1.write(1, 2, 33.3)
sh1.write(1, 3, 95371)

sh1.write(2, 0, '赤狐先生')
sh1.write(2, 1, 194.23)
sh1.write(2, 2, 17.8)
sh1.write(2, 3, 79980)

sh1.write(3, 0, '除暴')
sh1.write(3, 1, 130.05)
sh1.write(3, 2, 11.8)
sh1.write(3, 3, 42457)
```

```
sh1.write(4, 0, '疯狂原始人 2')
sh1.write(4, 1, 120.72)
sh1.write(4, 2, 10.9)
sh1.write(4, 3, 40697)
wb.save('excel01.xls')
```

读取数据

```
# pip install xlrd
import xlrd

wb = xlrd.open_workbook('excel01.xls')
print(wb)
# 获取并打印 sheet 数量
print( "sheet 数量:", wb.nsheets)
# 获取并打印 sheet 名称
print( "sheet 名称:", wb.sheet_names())
# 根据 sheet 索引获取内容
sh1 = wb.sheet_by_index(0)
# 或者
# 也可根据 sheet 名称获取内容
# sh = wb.sheet_by_name('成绩')
# 获取并打印该 sheet 行数和列数
print( u"sheet %s 共 %d 行 %d 列" % (sh1.name, sh1.nrows, sh1.ncols))

# 获取并打印某个单元格的值
print( "第一行第二列的值为:", sh1.cell_value(0, 1))
print( "第一行第二列的值为:", sh1.cell(0,1).value)

print( "第一行第二列的值为:", sh1.row(1)[2].value)


# 获取整行或整列的值

rows = sh1.row_values(0) # 获取第一行内容

cols = sh1.col_values(1) # 获取第二列内容


# 打印获取的行列值

print( "第一行的值为:", rows)

print( "第二列的值为:", cols)


# 获取单元格内容的数据类型

print( "第二行第一列的值类型为:", sh1.cell(1, 0).ctype)


# 遍历所有表单内容

for sh in wb.sheets():

    for r in range(sh.nrows):

        # 输出指定行

        print( sh.row(r))
```

更新数据

```
# pip insatll xlutils

# 导入相应模块

import xlrd

from xlutils.copy import copy
```

```
# 打开 excel 文件
readbook = xlrd.open_workbook("excel01.xls")
# 复制一份
wb = copy(readbook)

# 选取第一个表单
sh1 = wb.get_sheet(0)

# 在第五行新增写入数据
sh1.write(5, 0, '保家卫国-抗美援朝')
sh1.write(5, 1, 59.84)
sh1.write(5, 2, 5.1)
sh1.write(5, 3, 488)

# 选取第二个表单
sh2 = wb.add_sheet('汇总')
sh3 = readbook.sheet_by_index(0)
count = 0
for i in range(1,sh3.nrows):
    num = sh3.cell_value(i,3)
    count += num

# 替换总成绩数据
sh2.write(1, 0, count)
# 保存
wb.save('excel01.xls')
```

设置样式

```
# 导入 xlwt 库
import xlwt

wb = xlwt.Workbook()
# 新增两个表单页
sh1 = wb.add_sheet('账单')

ft = xlwt.Font()
ft.name = '微软雅黑'
ft.colour_index = 2
# 字体大小，11 为字号，20 为衡量单位
ft.height = 20*11
# 字体加粗
ft.bold = False
# 下划线
ft.underline = True
# 斜体字
ft.italic = True


# 设置单元格对齐方式
alignment = xlwt.Alignment()
# 0x01(左端对齐)、0x02(水平方向上居中对齐)、0x03(右端对齐)
alignment.horz = 1
```

```
# 0x00(上端对)、 0x01(垂直方向上居中对齐)、0x02(底端对齐)
alignment.vert = 2
# 设置自动换行
alignment.wrap = 1


# 设置列宽，一个中文等于两个英文等于两个字符，11 为字符数，256 为衡量单位
sh1.col(2).width = 6 * 256
sh1.row(0).height_mismatch = True
sh1.row(0).height = 6 * 256

# 设置边框
borders = xlwt.Borders()
# 细实线:1，小粗实线:2，细虚线:3，中细虚线:4，大粗实线:5，双线:6，细点虚线:7
# 大粗虚线:8，细点划线:9，粗点划线:10，细双点划线:11，粗双点划线:12，斜点划线:13
borders.left = 1
borders.right = 2
borders.top = 3
borders.bottom = 4
borders.left_colour = 3
borders.right_colour = 2
borders.top_colour = 2
borders.bottom_colour = 4

# 设置背景颜色
pattern = xlwt.Pattern()
# 设置背景颜色的模式
```

```
pattern.pattern = xlwt.Pattern.SOLID_PATTERN

# 背景颜色

pattern.pattern_fore_colour = 5



sy = xlwt.XFStyle()

sy.font = ft

sy.alignment= alignment

sy.borders = borders

sy.pattern = pattern


sy2 = xlwt.easyxf('font: bold on,color-index 4; align: wrap on, vert centre, horiz center')

sy3 = xlwt.easyxf('font: bold on,color-index 4; border: left 1 ,right_colour 3,right 1')


sh1.write(0,0,'吕小布')

sh1.write(0,1,'吕小布',sy)

sh1.write(0,2,'吕小布、貂的蝉、刘的备')

sh1.write(3,3,'鲁班 7 忠',sy)

sh1.write(3,1,'鲁班 7 忠',sy3)

wb.save('excel02.xls')
```

1.2 数据汇总

```
import re

import xlrd

import xlwt

from xlutils.copy import copy
```

```
def read_data():
    wb = xlrd.open_workbook('./data/data01.xlsx')
    sh = wb.sheet_by_index(0)
    das = []
    fen_type={}
    for r in range(sh.nrows):
        count = sh.cell_value(r,3)*sh.cell_value(r,4)
        das.append(count)
        key = sh.cell_value(r,0)
        if fen_type.get(key):
            fen_type[key] = fen_type.get(key)+count
        else:
            fen_type[key] = count
    return das,fen_type


def save_simple(data,fen):
    wb = xlrd.open_workbook('./data/data01.xlsx')
    sh_t = wb.sheet_by_index(0)
    wb2 = copy(wb)
    sh = wb2.get_sheet(0)
    for r in range(sh_t.nrows):
        sh.write(r,sh_t.ncols,data[r])


    sh2 = wb2.add_sheet('汇总')
    for i,key in enumerate(fen.keys()):
        sh2.write(i,0,key)
        sh2.write(i,1,fen.get(key))
    wb2.save('./销售表/data01_t.xlsx')
```

```
if __name__ == "__main__":
    d,f = read_data()
    save_simple(d,f)
```

1.3 表格拆分

```
import xlrd
from xlutils.copy import copy
def get_data():
    data = {}
    wb = xlrd.open_workbook('data01.xlsx')
    sh = wb.sheet_by_index(0)

    for i in range(sh.nrows):
        key = sh.cell_value(i,0)
        d = {'name':sh.cell_value(i,2),'type':sh.cell_value(i,1),'price':sh.cell_value(i,3),'count':sh.cell_value(i,3)}
        if data.get(key):
            data[key].append(d)
        else:
            t = [d]
            data[key] = t
    return data


def create_data(data):
    wb = xlrd.open_workbook('data01.xlsx')
    wb2 = copy(wb)
    for key in data.keys():
```

```
                    tw = wb2.add_sheet(key)
                    for i,d in enumerate(data.get(key)):
                        tw.write(i,0,d.get('type'))
                        tw.write(i,1,d.get('name'))
                        tw.write(i,2,d.get('price'))
                        tw.write(i,3,d.get('count'))
            wb2.save('data01_t.xlsx')



    if __name__ == "__main__":
        d = get_data()
        create_data(d)
```

```
# 导入模块
# 读入文件
# 获取活动表
# 获取单元格
# 单元格内容管理
# 设置行与高
# 背景颜色设置
# 公式的写入与获取
# 批量读写数据
# numpy 与 pandas 管理数据
```

1.4 openpyxl 的使用

```
def new():
    # 创建
    from openpyxl import Workbook
    # 实例化
    wb = Workbook()
    # 激活 worksheet
    ws = wb.active
    # 方式一：插入到最后(default)
    ws1 = wb.create_sheet("Mysheet")
    # 方式二：插入到最开始的位置
    ws2 = wb.create_sheet("Mysheet", 0)
    # 保存
    wb.save('new.xlsx')
def open():
    # 打开已有
    from openpyxl import load_workbook
    wb = load_workbook('data01.xlsx')
    # 选择已有表
    # sheet 名称可以作为 key 进行索引
    ws1 = wb.active
    ws3 = wb["Sheet1"]
    ws4 = wb.get_sheet_by_name("Sheet1")
    print(ws1 is ws3 is ws4)


def show_sheet():
    # 查看表名
    from openpyxl import load_workbook
    wb = load_workbook('data01.xlsx')
    print(wb.sheetnames)
    for s in wb:
        print(s.title)


def get_one_value():
    # 获取值
    from openpyxl import load_workbook
    wb = load_workbook('data01.xlsx')
    ws = wb.active
    v1 = ws['B3']
    v2 = ws.cell(row=4,column=3)
    print(v1.value,v2.value)
def get_many_value():
    # 获取值
    from openpyxl import load_workbook
    wb = load_workbook('data01.xlsx')
    ws = wb.active
    # 通过切片
    cell_range = ws['A1':'C2']
    # 通过行(列)
    colC = ws['C']
    colCD = ws['C':'D']
    row10 = ws[10]
    row_range = ws[5:10]
    print(cell_range)
```

```
    print(colC)

    print(colCD)

    print(row10)

    print(row_range)

    # 通过指定范围(行 → 行) 3 行 3 列

    for row in ws.iter_rows(min_row=2,max_row=5, max_col=3):

        for cell in row:

            print(cell.value)

    # 通过指定范围(列 → 列)

    for row in ws.iter_rows(min_col=3, max_col=5,min_row=1, max_row=5):

        for cell in row:

            print(cell.value)

    # 遍历所有 方法一

    print(tuple(ws.rows))

    # 遍历所有 方法二

    print(tuple(ws.columns))


def get_many_value2():

        # 获取值

    from openpyxl import load_workbook

    wb = load_workbook('data01.xlsx')

    ws = wb.active

    # sheet.rows 为生成器，里面是每一行的数据，每一行又由一个 tuple 包裹。

    # sheet.columns 类似，不过里面是每个 tuple 是每一列的单元格。

    # 因为按行，所以返回 A1, B1, C1 这样的顺序

    for row in ws.rows:

        for cell in row:
```

```
                    print(cell.value)


        # A1, A2, A3 这样的顺序
        for column in ws.columns:
            for cell in column:
                print(cell.value)
    def get_num():
        #  获取值
        from openpyxl import load_workbook
        wb = load_workbook('data01.xlsx')
        ws = wb.active
        #  获得最大列和最大行
        print(ws.max_row)
        print(ws.max_column)


    def remove_sheet():
         #  获取值
        from openpyxl import load_workbook
        wb = load_workbook('data01.xlsx')
        wb.remove('Sheet1')
        del wb['Sheet1']


    def set_value_style():
        from openpyxl.styles import Font, colors, Alignment
        bold_itatic_24_font = Font(name='微软雅黑', size=34, italic=True, color=colors.BLUE,
bold=True)


        # 创建
        from openpyxl import Workbook
        # 实例化
        wb = Workbook()
        # 激活 worksheet
        ws = wb.active
        # 方式一：插入到最后(default)
        ws = wb.create_sheet("Mysheet")
        ws['B2'] = 'Hello!'
        ws['B2'].font = bold_itatic_24_font
        # 保存
        wb.save('new2.xlsx')
def set_value():
         # 创建
        from openpyxl import Workbook
        # 实例化
        wb = Workbook()
        # 激活 worksheet
        ws = wb.active
        data = ['养老','医疗','公积金']
        for i,d in enumerate(data):
                ws.cell(i+1,1).value = d
        wb.save('new3.xlsx')
def set_style2():
        from openpyxl.styles import Font, colors, Alignment
       # 创建
        from openpyxl import Workbook
        # 实例化
        wb = Workbook()
        # 激活 worksheet
        ws = wb.active
        ws.row_dimensions[1].height = 40
        ws.column_dimensions['C'].width = 40
        data = ['养老','医疗','公积金']
        for i,d in enumerate(data):
            ws.cell(i+1,1).value = d
            ws.cell(i+1,1).alignment = Alignment(horizontal='center', vertical='center')

        wb.save('new4.xlsx')
def set_value2():
      # 创建
        from openpyxl import Workbook
        from datetime import date
        # 实例化
        wb = Workbook()
        # 激活 worksheet
        ws = wb.active
        rows = [
            ['Date', 'Batch 1', 'Batch 2', 'Batch 3'],
            [date(2020,12, 1), 40, 30, 25],
            [date(2020,12, 2), 40, 25, 30],
            [date(2020,12, 3), 50, 30, 45],
            [date(2020,12, 4), 30, 25, 40],
            [date(2020,12, 5), 25, 35, 30],
            [date(2020,12, 6), 20, 40, 35],
        ]
        for row in rows:
            ws.append(row)
        wb.save('new6.xlsx')

def set_merge():
        from openpyxl.styles import    Alignment
        # 创建
        from openpyxl import Workbook
        # 实例化
        wb = Workbook()
        # 激活 worksheet
        ws = wb.active
        ws.merge_cells('A1:c1')
        ws.merge_cells('D2:E5')
        ws['A1'] = '横向合并'
        ws.cell(1,1).alignment = Alignment(horizontal='center', vertical='center')

        ws['D2'] = '综合合并'
        ws['D2'].alignment = Alignment(horizontal='center', vertical='center')
        wb.save('new5.xlsx')
        # sheet.unmerge_cells('A1:C3')

def set_img():
         # 创建
        from openpyxl.chart import LineChart,Reference
from openpyxl import Workbook
from datetime import date
# 实例化
wb = Workbook()
# 激活 worksheet
ws = wb.active
rows = [
    ['Date', 'Batch 1', 'Batch 2', 'Batch 3'],
    [date(2020,12, 1), 40, 30, 25],
    [date(2020,12, 2), 40, 25, 30],
    [date(2020,12, 3), 50, 30, 45],
    [date(2020,12, 4), 30, 25, 40],
    [date(2020,12, 5), 25, 35, 30],
    [date(2020,12, 6), 20, 40, 35],
]
for row in rows:
    ws.append(row)

c1 = LineChart()
c1.title = "Line Chart"
# c1.style = 2
c1.y_axis.title = 'Size'
c1.x_axis.title = 'Test Number'
# min_row 第几行进行分类  min_col 第几列开始  max_col 取到第几列数据 max_row 取到第几行数据
data = Reference(ws, min_col=2, min_row=1, max_col=4, max_row=7)
# titles_from_data 是否启用标题
c1.add_data(data, titles_from_data=True)
```

```
        ws.add_chart(c1,'A9')
        wb.save('new7.xlsx')

def set_img2():
        from openpyxl.chart import PieChart,Reference
        from openpyxl import Workbook
        from openpyxl.chart.series import DataPoint

        data = [
                ['名称', '数值'],
                ['苹果', 50],
                ['草莓', 30],
                ['椰子', 10],
                ['荔枝', 40],
        ]
        wb = Workbook()
        ws = wb.active

        for row in data:
                ws.append(row)

        pie = PieChart()
        labels = Reference(ws, min_col=1, min_row=2, max_row=5)
        data = Reference(ws, min_col=2, min_row=1, max_row=5)
        pie.add_data(data, titles_from_data=True)
        pie.set_categories(labels)
        pie.title = "Pies sold by category"
        # Cut the first slice out of the pie

        # slice = DataPoint(idx=0, explosion=20)

        # pie.series[0].data_points = [slice]



        ws.add_chart(pie, "D1")

        wb.save('new8.xlsx')

def set_img3():

        from openpyxl import Workbook

        from openpyxl.chart import BarChart, Series, Reference



        wb = Workbook()

        ws = wb.active



        rows = [

                ('Number', 'Batch 1', 'Batch 2'),

                (2, 10, 30),

                (3, 40, 60),

                (4, 50, 70),

                (5, 20, 10),

                (6, 10, 40),

                (7, 50, 30),

        ]



        for row in rows:

                ws.append(row)

                chart1 = BarChart()

                chart1.type = "col"

                chart1.style = 10

                chart1.title = "Bar Chart"
```

```
    chart1.y_axis.title = 'Test number'
    chart1.x_axis.title = 'Sample length (mm)'

    data = Reference(ws, min_col=2, min_row=1, max_row=7, max_col=3)
    cats = Reference(ws, min_col=1, min_row=2, max_row=7)
    chart1.add_data(data, titles_from_data=True)
    chart1.set_categories(cats)
    chart1.shape = 4
    ws.add_chart(chart1, "A10")
    wb.save('new9.xlsx')
```

1.5 合并多个工作薄中一个工作表

```
# 打开已有
from openpyxl import load_workbook,Workbook
import os

def copy_data():
    wb = Workbook()
    ws = wb.active
    ta = []
    for i in os.listdir('./销售表'):
        tb = load_workbook(f'./销售表/{i}')
        ts = tb.active
        for x in range(1,ts.max_row):
            td = []
            for y in range(1,ts.max_column):
                d = ts.cell(x,y).value
                td.append(d)
            if td not in ta:
                ta.append(td)
    for a in ta:
        ws.append(a)
    wb.save('new10.xlsx')
if __name__ == "__main__":
    copy_data()
```

1.6 合并多个工作薄中所有工作表

```
# 打开已有
from openpyxl import load_workbook,Workbook
import os

def copy_data():
        wb = Workbook()
        for i in os.listdir('./销售表'):
            tb = load_workbook(f'./销售表/{i}')
            sheet_name = tb.sheetnames[0]
            print(sheet_name)
            ts = tb.active
            ta = wb.create_sheet(sheet_name)
            for x in range(1,ts.max_row):
                td = []
                for y in range(1,ts.max_column):
                    d = ts.cell(x,y).value
                    td.append(d)
                ta.append(td)
        del wb['Sheet']
        wb.save('new11.xlsx')

if __name__ == "__main__":

        copy_data()
```

1.7 文件名快速整理到 excel

1.8 一键格行换色

```
# 创建

from openpyxl import Workbook

from datetime import date

from openpyxl.styles import PatternFill

# 实例化

wb = Workbook()

# 激活 worksheet

ws = wb.active

rows = [

        ['Date', 'Batch 1', 'Batch 2', 'Batch 3'],

        [date(2020,12, 1), 40, 30, 25],

        [date(2020,12, 2), 40, 25, 30],

        [date(2020,12, 3), 50, 30, 45],

        [date(2020,12, 4), 30, 25, 40],

        [date(2020,12, 5), 25, 35, 30],

        [date(2020,12, 6), 20, 40, 35],

]

for row in rows:

        ws.append(row)


fille = PatternFill('solid',fgColor='fff000')
```

```
for i in range(1,ws.max_row+1): #遍历行号
    if i%2==0:   #如果行号除于 2 余数为 0，即为偶数时
        for j in range(1,ws.max_column+1): #  遍历当前行的所有表格
            ws.cell(i,j).fill=fille   #将当前行的每一个表格填充颜色
    i=i+1 #遍历下一行

wb.save('new12.xlsx')
```

1.9 快速生成工资条

```
from openpyxl import load_workbook,Workbook


wb = load_workbook('工资数据.xlsx')
ws = wb.active
title = ['工号','姓名','部门','基本工资','提成','加班工资','社保扣除','考勤扣除','应发工资']
for i ,row in enumerate(ws.rows):
    if i == 0:
        continue
    tb = Workbook()
    ts = tb.active
    ts.append(title)
    td = [cell.value for cell in row]
    ts.append(td)
    tb.save(f'./工资/{td[1]}.xlsx')
```

1.10 快速统计加班时间

```
# 创建
from openpyxl import Workbook,load_workbook
from datetime import date


from openpyxl.xml.constants import MAX_COLUMN


def create_data():
    # 实例化
    wb = Workbook()
    # 激活 worksheet
    ws = wb.active
    rows = [
        ['日期','姓名', '打卡时间'],
        [date(2020,12,1),'吕小布', '18:50'],
        [date(2020,12,2),'貂的蝉', '18:55'],
        [date(2020,12,3),'刘备', '19:50'],
        [date(2020,12,2),'吕小布', '20:10'],
        [date(2020,12,3),'吕小布', '19:30'],
    ]
    for row in rows:
        ws.append(row)
    wb.save('new13.xlsx')


def statistics():
    wb = load_workbook('new13.xlsx')
    ws = wb.active
    data = []
    for i in range(2,ws.max_row+1):
        t = []
```

```
            for j in range(1,ws.max_column+1):
                t.append(ws.cell(i,j).value)
            h,m = t[2].split(':')
            full =    int(h)*60+int(m)
            temp = full - 18*60
            t.append(temp)
            t[0] = t[0].date()
            data.append(t)
        wb2 = Workbook()
        ws2 = wb2.active
        for d in data:
            ws2.append(d)
        wb2.save('new14.xlsx')
    if __name__ == "__main__":
        statistics()
```

1.11 快速查找重复数据

```
# 创建
from openpyxl import Workbook,load_workbook
from datetime import date

from openpyxl.xml.constants import MAX_COLUMN

def create_data():
    # 实例化
    wb = Workbook()
    # 激活 worksheet
    ws = wb.active
```

```
        rows = [
            ['日期','姓名', '打卡时间'],
            [date(2020,12,1),'吕小布', '18:50'],
            [date(2020,12,2),'貂的蝉', '18:55'],
            [date(2020,12,3),'刘备', '19:50'],
            [date(2020,12,2),'吕小布', '20:10'],
            [date(2020,12,3),'吕小布', '19:30'],
        ]
        for row in rows:
            ws.append(row)
        wb.save('new13.xlsx')

    def statistics():
        wb = load_workbook('new13.xlsx')
        ws = wb.active
        data = []
        for i in range(2,ws.max_row+1):
            t = []
            for j in range(1,ws.max_column+1):
                t.append(ws.cell(i,j).value)
            h,m = t[2].split(':')
            full =   int(h)*60+int(m)
            temp = full - 18*60
            t.append(temp)
            t[0] = t[0].date()
            data.append(t)
        wb2 = Workbook()
        ws2 = wb2.active
        for d in data:
            ws2.append(d)
        wb2.save('new14.xlsx')
    if __name__ == "__main__":
        statistics()
```

第二章：《快速处理文档、排版》Word+PPT 自动化处理

2.1 Word 基本操作

```
def first():
    # 导入库
    from docx import Document
    # 新建空白文档
    doc1 = Document()
    # 新增文档标题
    doc1.add_heading('如何使用 Python 创建和操作 Word',0)
    # 保存文件
    doc1.save('word1.docx')
```

2.2 插入数据

```
def start1():
     # 导入库
    from docx import Document
    # 新建空白文档
    doc1 = Document()
    # 新增文档标题
    doc1.add_heading('如何使用 Python 创建和操作 Word',0)

    doc1.add_paragraph('此处 Tetle 信息','Title')

    # 创建一级标题

    doc1.add_heading('安装 python-docx 库',level = 1)

    # 创建二级标题

    doc1.add_heading('第一步：安装 Python',2)

    # 创建三级标题

    doc1.add_heading('第二步：安装 python-docx 库',3)

    # 创建段落描述

    paragraph = doc1.add_paragraph('''

     Word 文档在我们现在的生活和工作中都用的比较多，我们平时都使用 wps 或者
office 来对 Word 进行处理，

     可能没想过它可以用 Python 生成，下面我们就介绍具体如何操作……''')

     # 段落中增加文字

    run = paragraph.add_run('(注意：这个段落里的数据')


    paragraph = doc1.add_paragraph('''

     Word 文档在我们现在的生活和工作中都用的比较多，我们平时都使用 wps 或者
office 来对 Word 进行处理，

     可能没想过它可以用 Python 生成，下面我们就介绍具体如何操作……''')

     # 段落中增加文字

    run = paragraph.add_run('(注意：这个段落里的数据')


    doc1.add_paragraph('哪个不是动物：')

    # 增加无序列表
```

```
doc1.add_paragraph('苹果', style='List Bullet')

doc1.add_paragraph('喜洋洋', style='List Bullet')

doc1.add_paragraph('懒洋洋', style='List Bullet')

doc1.add_paragraph('沸洋洋', style='List Bullet')

doc1.add_paragraph('灰太狼', style='List Bullet')

doc1.add_paragraph('2020 年度计划：')

#  增加有序列表

doc1.add_paragraph('CSDN 达到博客专家', style='List Number')

doc1.add_paragraph('每周健身三天', style='List Number')

doc1.add_paragraph('每天学习一个新知识点', style='List Number')

doc1.add_paragraph('学习 50 本书', style='List Number')

doc1.add_paragraph('减少加班时间', style='List Number')


 #  增加引用

doc1.add_paragraph('这里是我们引用的一段话：用 Python 改变人生，改变世界，
FIGHTING。', style='Intense Quote')

#   增加图片  默认情况下，添加的图像以原始大小显示

picture = doc1.add_picture('aaa.jpg',width = Inches(5.0))

# height = picture.height

# width    = picture.width

# sc = (doc1.sections[0].page_width/10-doc1.sections[0].left_margin/10*2)/(width/10)

# picture.height = int(height * sc)

# picture.width = int(width*sc)+100


# 增加表格，这是表格头
```

```
    table = doc1.add_table(rows=1, cols=3)

    hdr_cells = table.rows[0].cells

    hdr_cells[0].text = '编号'

    hdr_cells[1].text = '姓名'

    hdr_cells[2].text = '职业'


    # 这是表格数据

    records = (

        (1, '张三', 'Python'),

        (2, '张五', 'Java'),

        (3, '马六', 'JavaScript'),

        (4, '李四', 'C++')

    )


    # 遍历数据并展示

    for id, name, work in records:

        row_cells = table.add_row().cells

        row_cells[0].text = str(id)

        row_cells[1].text = name

        row_cells[2].text = work


    # 保存文件

    doc1.save('word2.docx')
```

2.3 修改样式

```
def start2():
```

```
# 样式处理
  # 导入库
from docx import Document
# 新建空白文档
doc1 = Document()
from docx.enum.text import WD_ALIGN_PARAGRAPH #设置对象居中、对齐等。
from docx.shared import Inches #设置大小英寸
from docx.shared import Pt #设置像素、缩进等
from docx.shared import RGBColor #设置字体颜色
# doc1.add_paragraph('苹果', style='List Bullet')
p1 = doc1.add_paragraph('这是段落 1：\n')
p1.add_run('''这是内容！！ 1\n''').bold = True
p1.add_run('''这是内容！！ 2\n''').italic = True #  斜体
info = p1.add_run('''这是内容！！ 3\n''')
info.font.size = Pt(20)
info.font.color.rgb = RGBColor(255,0,0)


# 设置居中、左右对齐、缩进、制表符
doc1.add_paragraph(''' 这 是 段 落 3 ： 居 中 \n''').paragraph_format.alignment = WD_ALIGN_PARAGRAPH.CENTER #  居中
# 默认 Inches(0.5)等于四个空格
p = doc1.add_paragraph('''这是段落 4：这个内容有点多，必须要换行啊！！！！！！dafjoiajf 葛晓洁 jaofj 鞯介 oiwjfowaijewfoaijw 恶法\n''').paragraph_format.left_indent=Inches(0.5) #  整段缩进
p = doc1.add_paragraph('''这是段落 5：这个内容有点多，必须要换行啊！！！！！！dafjoiajf 葛 晓 洁  jaofj 鞯 介 oiwjfowaijewfoaijw 恶 法 ''').paragraph_format.first_line_indent =
Inches(0.25)

        # 设置段落间距 分为段落前 和 段落后

        doc1.add_paragraph('''这是段落 6： 段后距离\n''').paragraph_format.space_after = Pt(0)

        doc1.add_paragraph('''这是段落 7：段前距离\n''').paragraph_format.space_before = Pt(0)


        p2 = doc1.add_paragraph('''这是段落 2： \n''')

        p2.paragraph_format.line_spacing = 1.5     #     设置行间距

        p2.add_run('''这是内容！ ！ 1\n''')

        p2.add_run('''这是内容！ ！ 2\n''')

         # 新建空白文档

        doc1 = Document()

        # 设置英文字体

        p1 = doc1.add_paragraph('这里设置英文字体： ')

        run = p1.add_run('This Font is Times New 11 ')

        run.font.name = '微软雅黑'

        # 设置中文字体

        p2 = doc1.add_paragraph('这里设置中文字体： ')

        run2 = p2.add_run('当前字体为黑体 111')

        run2.font.name='.'

        r = run._element

        r.rPr.rFonts.set(qn('w:eastAsia'), '黑体')

        # 保存文件

        doc1.save('word4.docx')
```

2.4 生成文件

```
 # 导入库
from docx import Document
from docx.shared import Pt #设置像素、缩进等
from docx.shared import RGBColor #设置字体颜色
from docx.enum.text import WD_ALIGN_PARAGRAPH
from docx.shared import Inches #设置图像大小
from docx.shared import RGBColor #设置字体颜色
from docx.oxml.ns import qn

card = '00001'
year = 2020
month = 12
date = 12
hour = 12
minute = 1
info = '闯红灯'
money = 300

# 新建空白文档
doc1 = Document()
title = doc1.add_paragraph()
p = title.add_run('车辆违章处罚通知书')
p.font.size = Pt(30)
p.font.color.rgb=RGBColor(255,0,0)
p.font.name='微软雅黑'
```

```
    r = p._element
    r = p._element.rPr.rFonts.set(qn('w:eastAsia'), '黑体')
    title.paragraph_format.alignment = WD_ALIGN_PARAGRAPH.CENTER


    content = doc1.add_paragraph()
    content.paragraph_format.first_line_indent = Inches(0.25)
    content.add_run(f'''  京   {card}   车 于   {year}   年
{month} 月  {date} 日  {hour} 时  {minute} 分在营运过程中出现  {info}  (违章)现象,
公司按照安全法规和公司相关制度规定决定对该车驾驶员处以 {money} 元罚款,要求你在今
后的营运过程中严格按照相关法律法规运行。(注:罚款金额请在返程后立即到公司缴纳)
                驾驶员签字:              年 月 日


    ''')
    doc1.save('word5.docx')
```

2.5 读取文件

```
    import docx


    doc = docx.Document('word2.docx')
    for p in doc.paragraphs:
        print(p.text)
    for t in doc.tables:
        for r in t.rows:
            for c in r.cells:
                print(c.text)
```

2.6 模板生成文件

```
    import docx
    import os
```

```
infos = [
    ['00001',2020,12,12,13,30,'闯红灯',300],
    ['00002',2020,12,12,13,30,'闯红灯',300],
    ['00003',2020,12,12,13,30,'闯红灯',300],
    ['00004',2020,12,12,13,30,'闯红灯',300],
]
for info in infos:
    doc = docx.Document('word_模板.docx')
    for p in doc.paragraphs:
        for run in p.runs:
            run.text = run.text.replace('{0}',info[0])
            run.text = run.text.replace('{1}',str(info[1]))
            run.text = run.text.replace('{2}',str(info[2]))
            run.text = run.text.replace('{3}',str(info[3]))
            run.text = run.text.replace('{4}',str(info[4]))
            run.text = run.text.replace('{5}',str(info[5]))
            run.text = run.text.replace('{6}',info[6])
            run.text = run.text.replace('{7}',str(info[7]))

    if not os.path.exists('./生成 word'):
        os.makedirs('./生成 word')
    doc.save(f'./生成 word/车辆_{info[0]}.docx')
```

2.7 批量生成 word 合成文件

```
from win32com.client import gencache
from win32com.client import constants, gencache
def createPdf(wordPath, pdfPath):
        """

        word 转 pdf

        :param wordPath: word 文件路径

        :param pdfPath:   生成 pdf 文件路径

        """

        word = gencache.EnsureDispatch('Word.Application')

        doc = word.Documents.Open(wordPath, ReadOnly=1)

        doc.ExportAsFixedFormat(pdfPath,

                                constants.wdExportFormatPDF,

                                Item=constants.wdExportDocumentWithMarkup,

CreateBookmarks=constants.wdExportCreateHeadingBookmarks)

        word.Quit(constants.wdDoNotSaveChanges)


    if __name__ == "__main__":

        createPdf('word2.docx','pdf1.pdf')
```

2.8 word 转 PDF

```
    from win32com.client import gencache

    from win32com.client import constants, gencache

    def createPdf(wordPath, pdfPath):

        """

        word 转 pdf

        :param wordPath: word 文件路径

        :param pdfPath:   生成 pdf 文件路径

        """

        word = gencache.EnsureDispatch('Word.Application')

        doc = word.Documents.Open(wordPath, ReadOnly=1)
```

```
            doc.ExportAsFixedFormat(pdfPath,
                                                    constants.wdExportFormatPDF,
                                                    Item=constants.wdExportDocumentWithMarkup,

CreateBookmarks=constants.wdExportCreateHeadingBookmarks)
            word.Quit(constants.wdDoNotSaveChanges)


      if __name__ == "__main__":
            createPdf('word2.docx','pdf1.pdf')
```

2.9 读取 PDF

```
def extract_information(pdf_path):
      # pip install pypdf3
      from PyPDF3 import PdfFileReader
      pdf = None
      with open(pdf_path,'rb') as f:
            pdf = PdfFileReader(f)
            information = pdf.getDocumentInfo()
            number_of_pages = pdf.getNumPages()
            for i in range(number_of_pages):
                  print(pdf.getPage(i).extractText())
      txt = f"""
      Information about {pdf_path}:


      Author:{information.author}
      Creator:{information.creator}
      Producer:{information.producer}
      Subject:{information.subject}
```

```
    Title:{information.title}

    Number of pages:{number_of_pages}

    """


    print(txt)


    return information


def read_pdf(path):

    # pip install pdfplumber

    import pdfplumber

    with pdfplumber.open(path) as pdf:

        #len(pdf.pages)为 PDF 文档页数

        for i in range(1):

        #pdf.pages[i] 是读取 PDF 文档第 i+1 页

            page = pdf.pages[i]

            #page.extract_text()函数即读取文本内容，下面这步是去掉文档最下面
的页码

            print(page.extract_text())
```

2.10 合并 PDF

```
def merger_pdf(path1,path2):

    from PyPDF3 import PdfFileReader, PdfFileWriter

    writer = PdfFileWriter()

    for p in [path1,path2]:

        tp = PdfFileReader(open(p,'rb'))
```

```
            for page in range(tp.getNumPages()):
                writer.addPage(tp.getPage(page))
        with open('merge.pdf', 'wb') as out:
            writer.write(out)
```

2.11 拆分 PDF

```
    def chai_pdf(path):
        from PyPDF2 import PdfFileReader, PdfFileWriter


        p = PdfFileReader(open(path,'rb'))
        for page in range(p.getNumPages()):
            writer = PdfFileWriter()
            writer.addPage(p.getPage(page))
            with open(f'./生成 pdf/chai_{page}.pdf', 'wb') as out:
                writer.write(out)
```

2.12 加密 PDF

```
def jia_mi(path):
    from PyPDF2 import PdfFileReader, PdfFileWriter


    p = PdfFileReader(open(path,'rb'))
    writer = PdfFileWriter()
    writer.encrypt('123')
    for page in range(p.getNumPages()):
        writer.addPage(p.getPage(page))


    with open(f'./生成 pdf/jiami.pdf', 'wb') as out:
        writer.write(out)
```

2.13 PPT 基本操作

slide（幻灯片）：一个 PPT 由一系列 slide 构成。

slide_master（幻灯片母版）：母版可定义主题样式基准。

slide_layouts（模版）：创建幻灯片时可选择的模版。

shape（形状）：包含一切可视元素，通过 slide.shapes 可访问 slide 内元素。

placeholder（占位符）：在模板中占据位置，如图片、文字等。

paragraph（段落）：文本段，可以直接设置整段文本样式。

```
def base_use1():
    from pptx import Presentation
    ppt = Presentation()
    # 创建标题页
    title_slide_layout = ppt.slide_layouts[0]
    ppt.slides.add_slide(title_slide_layout)

    title_slide_layout = ppt.slide_layouts[3]
    ppt.slides.add_slide(title_slide_layout)

    # 用来填充背景
    title_slide_layout = ppt.slide_layouts[1]
    slide = ppt.slides.add_slide(title_slide_layout)

    ppt.save('ppt2.pptx')
def base_use2():
    from pptx import Presentation
    from pptx.util import Inches,Pt

    ppt = Presentation()

      # 创建标题页

    slide = ppt.slides.add_slide(ppt.slide_layouts[1])

    # 获取内容框

    shapes = slide.shapes

    title = shapes.title

    title.text = 'This is Python title~'

    content = shapes.placeholders[1]

    content.text = 'Content Info' #    等同于   content.text_frame.text='first Content'


    # 创建标题页 - 增加列表

    slide2 = ppt.slides.add_slide(ppt.slide_layouts[1])

    # 获取内容框

    shapes2 = slide2.shapes

    content = shapes2.placeholders[1]

    tf = content.text_frame

    p = tf.add_paragraph()

    p.text ='这个是什么内容呢'

    p.level = 1

    p = tf.add_paragraph()

    p.text ='这个是什么内容呢'

    p.level = 2

    p = tf.add_paragraph()

    p.text = '文本框内增加一个粗体段落'

    p.font.bold = True

    p = tf.add_paragraph()
```

```
        p.text = '文本框内增加一个大字体段落'

        p.font.size = Pt(40)



        slide = ppt.slides.add_slide(ppt.slide_layouts[6])

        left = top = width = height = Pt(300)

        txt_box = slide.shapes.add_textbox(left, top, width, height)

        tf = txt_box.text_frame

        tf.text = '文本框内容'

        p = tf.add_paragraph()

        p.text = '文本框内增加一个粗体段落'

        p.font.bold = True

        p = tf.add_paragraph()

        p.text = '文本框内增加一个大字体段落'

        p.font.size = Pt(40)


        ppt.save('ppt3.pptx')
```

2.14 PPT 增加图片

```
    def base_use3():

        from pptx import Presentation

        from pptx.util import Inches, Pt

        from pptx.enum.shapes import MSO_SHAPE


        ppt = Presentation()

        #  创建标题页

        slide = ppt.slides.add_slide(ppt.slide_layouts[1])
        # 获取内容框
        shapes = slide.shapes
        # 增加两张图片，一大一小
        left = top = Pt(30)
        pic = shapes.add_picture('aaa.jpg', left, top)

        left = Inches(6)
        height = Inches(3)
        pic = slide.shapes.add_picture('aaa.jpg', left, top, height=height)
        ppt.save('ppt3.pptx')
```

2.15 PPT 增加流程图

```
    def base_use4():
        from pptx import Presentation
        from pptx.util import Inches, Pt
        from pptx.enum.shapes import MSO_SHAPE

        ppt = Presentation()
        # 增加一个 AutoShape 流程图
        slide = ppt.slides.add_slide(ppt.slide_layouts[5])
        shapes = slide.shapes
        shapes.title.text = '流程图'
        left = Inches(0.93)   # 0.93" centers this overall set of shapes
        top = Inches(3.0)
        width = Inches(1.75)
        height = Inches(1.0)
        shape = shapes.add_shape(MSO_SHAPE.PENTAGON, left, top, width, height)
        shape.text = '第一步'
        left = left + width - Inches(0.4)

        #  第一个形状左边平，第二个开始，左边有尖角，需要更长

        width = Inches(2.0)

        for n in range(2, 6):

            shape = shapes.add_shape(MSO_SHAPE.CHEVRON, left, top, width, height)

            shape.text = f'第{n}步'

            left = left + width - Inches(0.4)


        ppt.save('ppt4.pptx')
```

2.16 放图表

```


    def base_use5():

        from pptx import Presentation

        from pptx.util import Inches, Pt,Cm

        from pptx.chart.data import CategoryChartData

        from pptx.enum.chart import XL_CHART_TYPE, XL_TICK_MARK, XL_LABEL_POSITION,
XL_LEGEND_POSITION


        prs = Presentation()

        blank_slide_layout = prs.slide_layouts[6]

        slide = prs.slides.add_slide(blank_slide_layout)

        #  增加一个图表

        chart_data = CategoryChartData()

        chart_data.categories = ['A 销售额', 'B 销售额', 'C 销售额']

        chart_data.add_series('Q1 销售', (100, 120, 200))

        chart_data.add_series('Q2 销售', (120, 150, 180))

        chart_data.add_series('Q3 销售', (150, 180, 120))
chart_data.add_series('Q4 销售', (130, 210, 150))

x, y, cx, cy = Inches(2), Inches(2), Inches(6), Inches(4)


chart = slide.shapes.add_chart(

        XL_CHART_TYPE.COLUMN_CLUSTERED, x, y, cx, cy, chart_data

).chart

chart.chart_style = 10

chart.font.size=Pt(12)

# 设置图表的轴信息

category_axis = chart.category_axis

# 是否有方网格线

category_axis.has_major_gridlines = True

# 是否有刻度

category_axis.minor_tick_mark = XL_TICK_MARK.OUTSIDE

# 分类名斜体

# category_axis.tick_labels.font.italic = True

# 分类名斜体

category_axis.tick_labels.font.size = Pt(12)


plot = chart.plots[0] # 用 Plot 对象访问标签

# 是否显示数据标签

plot.has_data_labels = True

# 获取数据标签控制类

data_labels = plot.data_labels

data_labels.position = XL_LABEL_POSITION.INSIDE_END

# data_labels.font.size=Pt(5)
# 是否显示图例
chart.has_legend = True
# 图例的位置
chart.legend.position = XL_LEGEND_POSITION.RIGHT
# 是否重叠数据
chart.legend.include_in_layout = False
chart.legend.font.size=Pt(15)


# 增加一个饼图
slide = prs.slides.add_slide(blank_slide_layout)
chart_data = CategoryChartData()
chart_data.categories = ['A 销售额', 'B 销售额', 'C 销售额']
chart_data.add_series('年度销售比', (0.276, 0.365, 0.359))
chart = slide.shapes.add_chart(
    XL_CHART_TYPE.PIE, x, y, cx, cy, chart_data
).chart
chart.has_legend = True
chart.legend.position = XL_LEGEND_POSITION.BOTTOM
chart.legend.include_in_layout = False
chart.plots[0].has_data_labels = True
data_labels = chart.plots[0].data_labels
data_labels.number_format = '0%'
data_labels.position = XL_LABEL_POSITION.OUTSIDE_END
prs.save('ppt5.pptx')
```

第三章：《机器人助理办公》邮件自动化处理

3.1 Python 发送普通邮件

```
# smtplib
# email
import smtplib
from email.mime.text import MIMEText    #  邮件正文
from email.header import Header    #  邮件头
from email.mime.multipart import MIMEMultipart
#  登陆邮箱
smtp_obj = smtplib.SMTP('smtp.qq.com')    #  邮箱发送服务器（ssL 485 报错去掉端口）
smtp_obj.login('398707160@qq.com', 'zibrmlljboscbiced')    #  邮箱用户名，密码(授权码)
mail_body_context = 'This is Test Email,你要的邮件来啦'
msg_body = MIMEText(mail_body_context, 'plain', 'utf-8')


# msg_body['From'] = Header('测试人事部', 'utf-8')    #  发送者
# msg_body['Subject'] = Header('三国公司 2020 年 5 月份工资条', 'utf-8')    #  主题


#  发邮件
smtp_obj.sendmail('398707160@qq.com', ['hotelmail@126.com'], msg_body.as_string())
```

3.2 Python 发送 html 邮件

```
# smtplib
# email
import smtplib
```

```
from email.mime.text import MIMEText    # 邮件正文

from email.header import Header    # 邮件头

from email.mime.multipart import MIMEMultipart

# 登陆邮箱

smtp_obj = smtplib.SMTP('smtp.qq.com')    # 邮箱发送服务器（ssL 485 报错去掉端口）

smtp_obj.login('398707160@qq.com', 'zibrmlljboscbice')    # 邮箱用户名，密码(授权码)

mail_body_context = """

<h1 style='color:red'>这个是个 HTML 标题</h1>

<p>Python  邮件发送测试...</p>

<p><a href="http://www.itbaizhan.cn">这是一个链接</a></p>

"""


msg_body = MIMEText(mail_body_context, 'html', 'utf-8')


msg_body['From'] = Header('测试人事部', 'utf-8')    # 发送者

msg_body['Subject'] = Header('三国公司 2020 年 5 月份工资条', 'utf-8')    # 主题


# 发邮件

smtp_obj.sendmail('398707160@qq.com', ['hotelmail@126.com'], msg_body.as_string())
```

3.3 Python 发送附件邮件

```
# smtplib

# email

import smtplib

from email.mime.text import MIMEText    # 邮件正文
from email.header import Header    # 邮件头

from email.mime.multipart import MIMEMultipart

from email.mime.application import MIMEApplication

# 登陆邮箱

smtp_obj = smtplib.SMTP('smtp.qq.com')    # 邮箱发送服务器（ssL 485 报错去掉端口）

smtp_obj.login('398707160@qq.com', 'zibrmlljboscbice')    # 邮箱用户名，密码(授权码)

mail_body_context = """

<h1 style='color:red'>这个是个 HTML 标题</h1>

<p>Python  邮件发送测试...</p>

<p><a href="http://www.itbaizhan.cn">这是一个链接</a></p>

"""

msg_body = MIMEMultipart()

msg_body.attach(MIMEText(mail_body_context, 'html', 'utf-8'))

msg_body['From'] = Header('测试人事部', 'utf-8')    # 发送者

msg_body['Subject'] = Header('三国公司 2020 年 5 月份工资条', 'utf-8')    # 主题


# 构造附件 1，传送当前目录下的 京东 601.pdf  文件

att1 = MIMEApplication(open('京东 601.pdf', 'rb').read())

att1.add_header('Content-Disposition','attachment',filename='aa.pdf')

msg_body.attach(att1)



# 发邮件

smtp_obj.sendmail('398707160@qq.com', ['hotelmail@126.com'], msg_body.as_string())
```

3.4 发送工资条

```
# 员工发工资的脚本
from openpyxl import load_workbook
import smtplib
from email.mime.text import MIMEText   # 邮件正文
from email.header import Header   # 邮件头


# 加载 excel 文件
wb = load_workbook('aaa.xlsx', data_only=True)


sheet1 = wb.active
# print(sheet1)


# 登陆邮箱
smtp_obj = smtplib.SMTP('smtp.qq.com')   # 邮箱发送服务器（ssL 485 报错去掉端口）
smtp_obj.login('398707160@qq.com', 'zibrmlljboscbice')   # 邮箱用户名，密码(授权码)


count = 0
table_col_html = '<tr>'   # 表头
for row in sheet1.iter_rows():
    # print(row[0].value,row[1].value)
    count += 1
    if count == 1:
        for col in row:
            table_col_html += f"<td>{col.value}</td>"
        table_col_html += '</tr>'
        continue
    else:
        row_test = '<tr>'   #  开始一行
        for cell in row:
            # print(cell.value,end=',')
            row_test += f"<td>{cell.value}</td>"
        row_test += "</tr>"   #  结束一行

        name = row[1].value
        staff_email = row[9].value
        print(staff_email,name)
    #     print(row_test)

        mail_body_context = f"""
          <h3>{name},你好：</h3>
          <p>请查收 2020 年 6 月的工资条。。。</p>
          <table border="1">
          {table_col_html}
          {row_test}
          </table>
        """
        # print(mail_body_context)
        msg_body = MIMEText(mail_body_context, 'html', 'utf-8')

        msg_body['From'] = Header('测试人事部', 'utf-8')   #  发送者
        # msg_body['To'] = staff_email
        # msg_body['To'] = Header(f'{staff_email}','utf-8') #  接受者
```

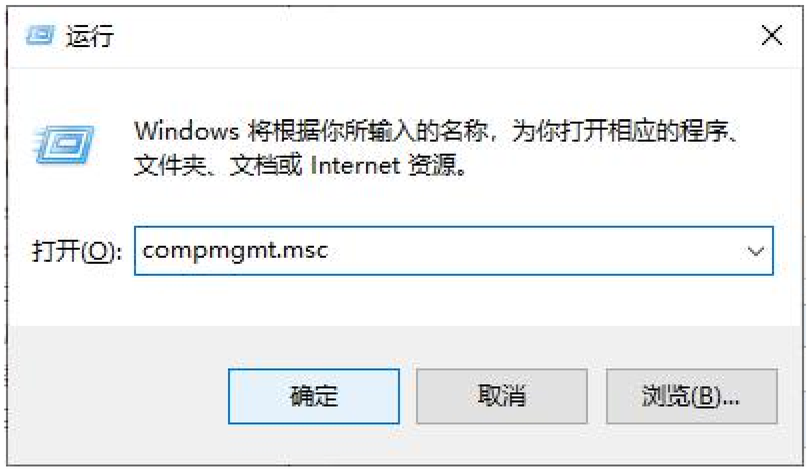
```
            msg_body['Subject'] = Header('三国公司 2020 年 5 月份工资条', 'utf-8')    # 主题

            # print(msg_body.as_string())

            # 发邮件

            smtp_obj.sendmail('398707160@qq.com',                    ['hotelmail@126.com'],
msg_body.as_string())

            print(f"成功发送工资条到:{staff_email}-{name}.....")
```

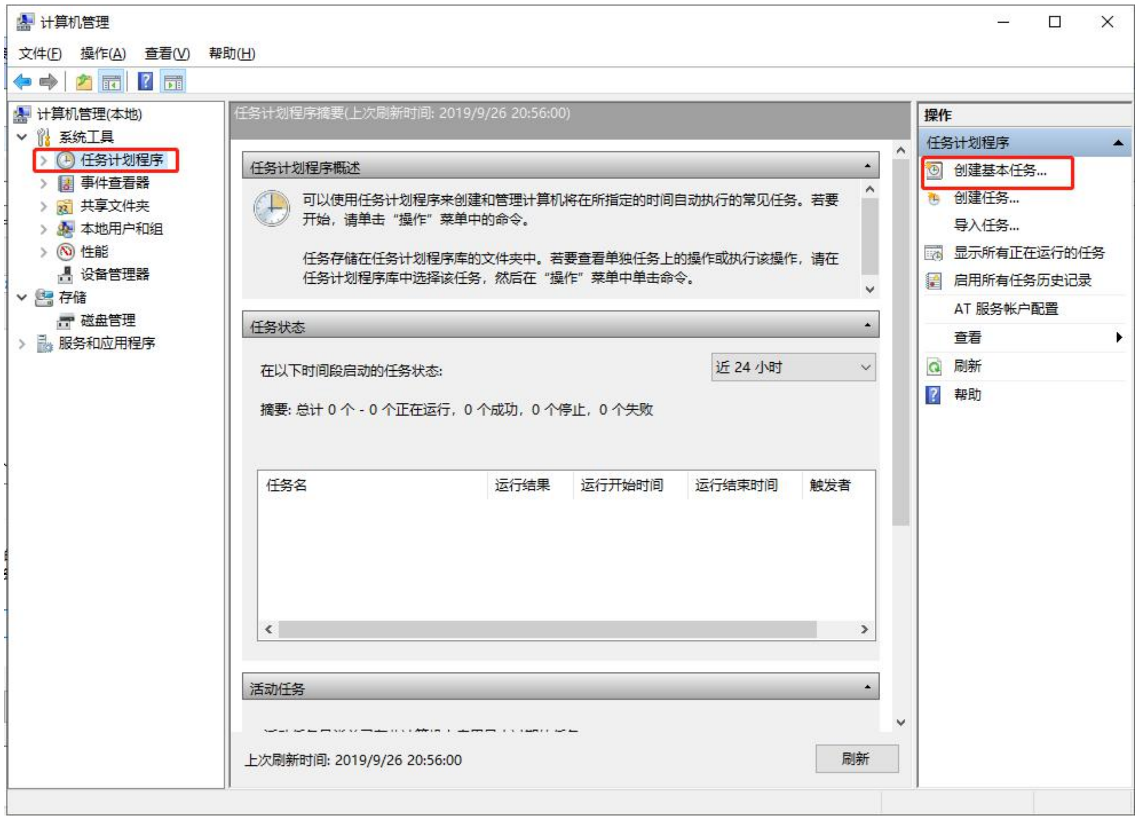
3.4 Python 中 zmail 模块

```
    # pip install zmail

    def mail01():

        import zmail

        server = zmail.server('398707160@qq.com', 'zibrmlljboscbice')


        mail = {

            'subject': '这个是我们的主题',

            'content_text': '这个是内容啊！！',

            'attachments': ['m1.xlsx','京东 601.pdf'],

            'from':'这个发送者<sdf>'

        }


        # 发送邮件

        server.send_mail('hotelmail@126.com',mail)

    def mail02():

        import zmail

        server = zmail.server('398707160@qq.com', 'zibrmlljboscbice')
        mail = {
                'subject': '这个是我们的主题',
                'content_html': ['<h1>这个是内容啊！！ </h1>'],
                'attachments': ['m1.xlsx','京东 601.pdf'],
        }
        server.send_mail('hotelmail@126.com',mail)
def mail03():
        import zmail
        server = zmail.server('398707160@qq.com', 'qxpokkjnpwolcafi')

        latest_mail = server.get_latest()
        # zmail.show(latest_mail)
        print(latest_mail.get('subject'))
        print(latest_mail.get('id'))
        print(latest_mail.get('from'))
        print(latest_mail.get('to'))
        print(latest_mail.get('content_text'))
        print(latest_mail.get('content_html'))
        print(latest_mail.get('date'))
```

3.5 windows 中定时任务

Windows 键+R，调出此窗口，输入 compmgmt.msc

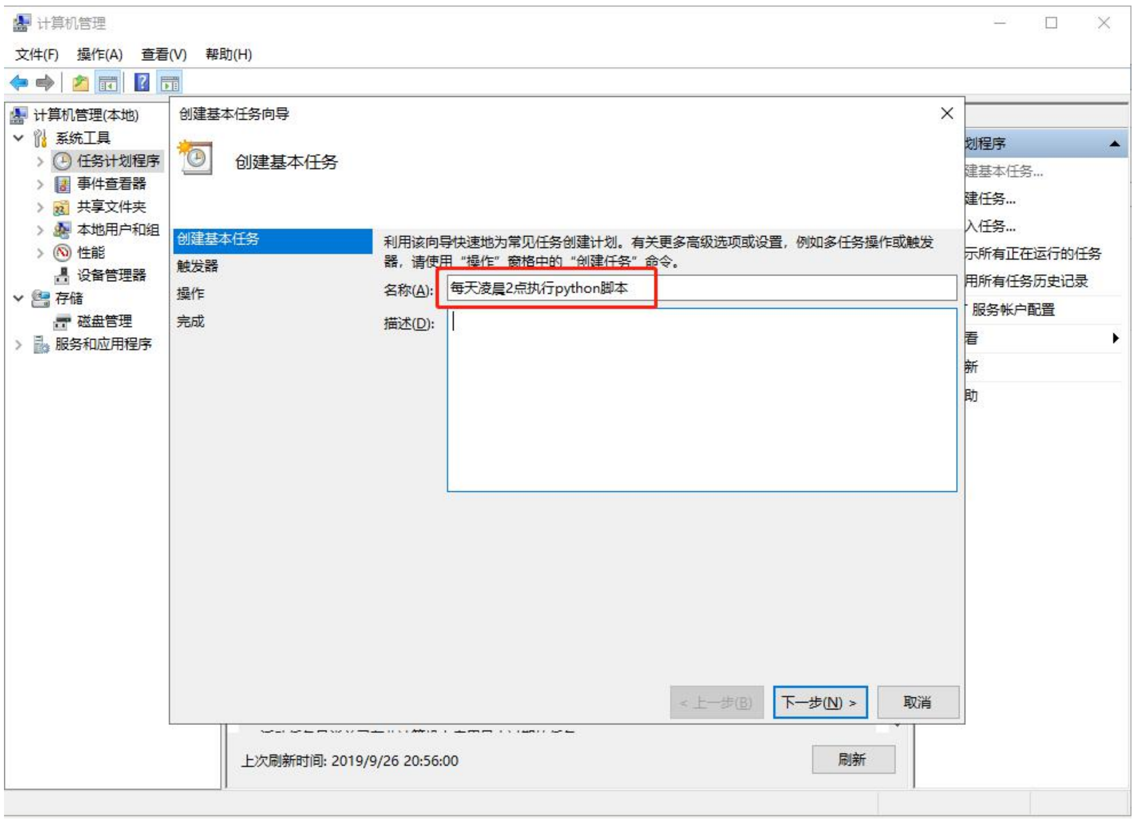

运行
Windows 将根据你所输入的名称，为你打开相应的程序、文件夹、文档或 Internet 资源。
打开(O): compmgmt.msc
确定
取消
浏览(B)...



可以使用任务计划程序来创建和管理计算机将在所指定的时间自动执行的常见任务。若要开始，请单击“操作”菜单中的命令。

任务存储在任务计划程序库的文件夹中。若要查看单独任务上的操作或执行该操作，请在任务计划程序库中选择该任务，然后在“操作”菜单中单击命令。

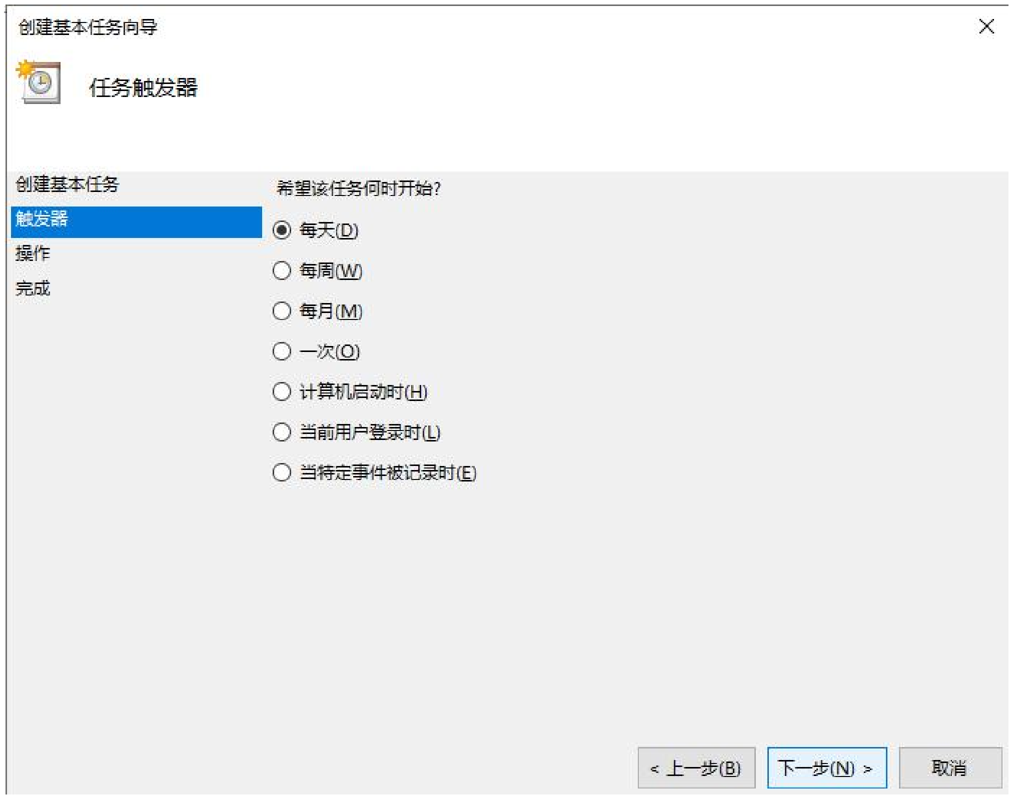
摘要: 总计 0 个 - 0 个正在运行，0 个成功，0 个停止，0 个失败



利用该向导快速地为常见任务创建计划。有关更多高级选项或设置，例如多任务操作或触发器，请使用"操作"窗格中的"创建任务"命令。

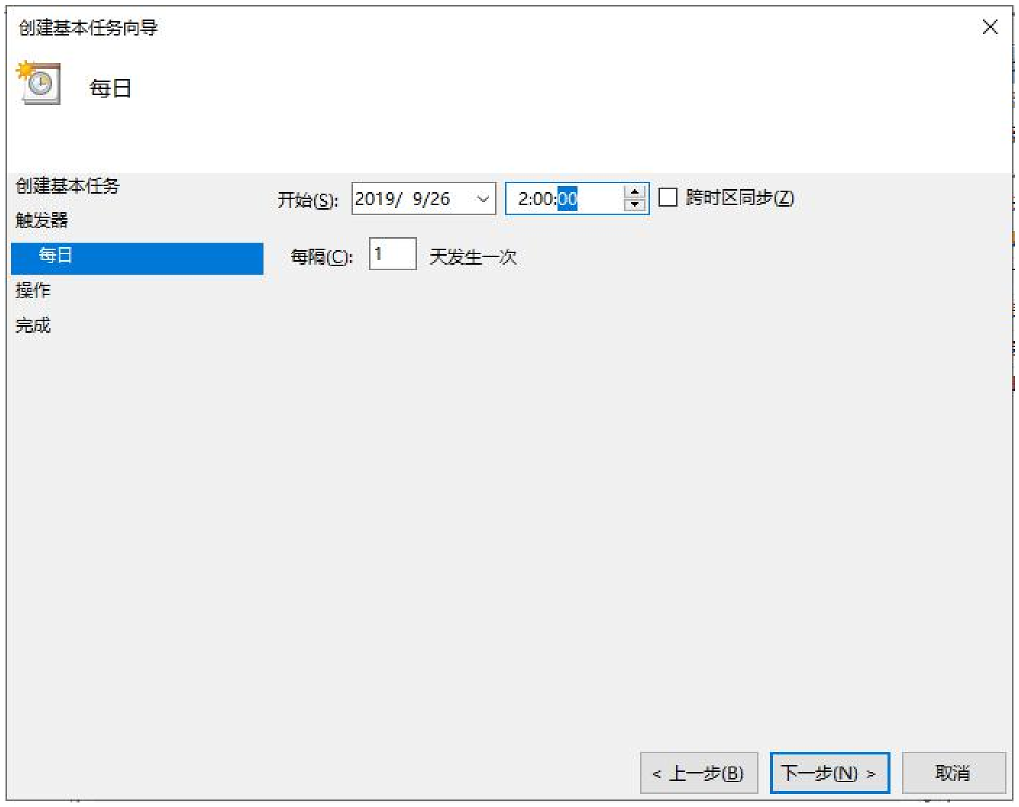
名称(A): 每天凌晨2点执行python脚本

描述(D):



希望该任务何时开始?

- [✓ 每天(D)]
- 每周(W)
- 每月(M)
- 一次(O)
- 计算机启动时(H)
- 当前用户登录时(L)
- 当特定事件被记录时(E)


创建基本任务向导

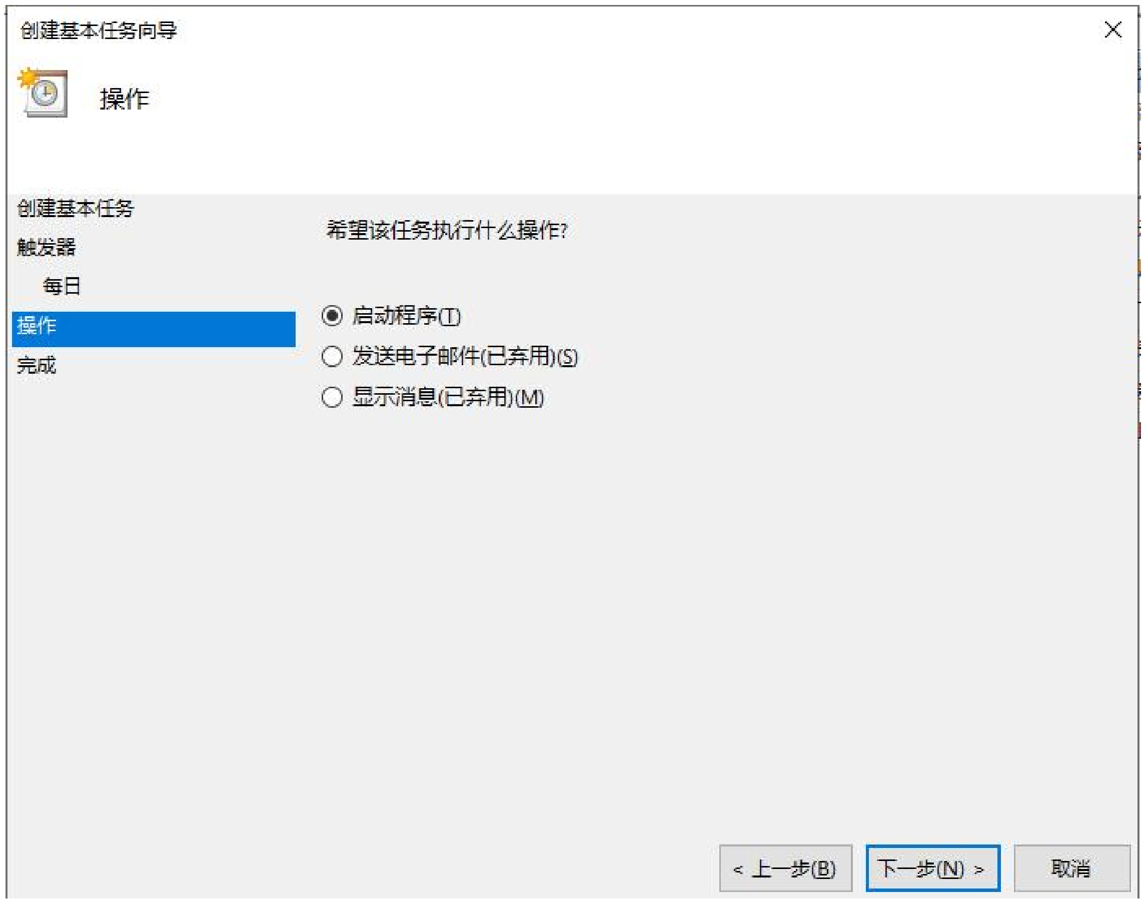
每日

创建基本任务
触发器
每日
操作
完成

开始(S): 2019/ 9/26 2:00:00 ☐ 跨时区同步(Z)

每隔(C): 1 天发生一次

< 上一步(B) 下一步(N) > 取消

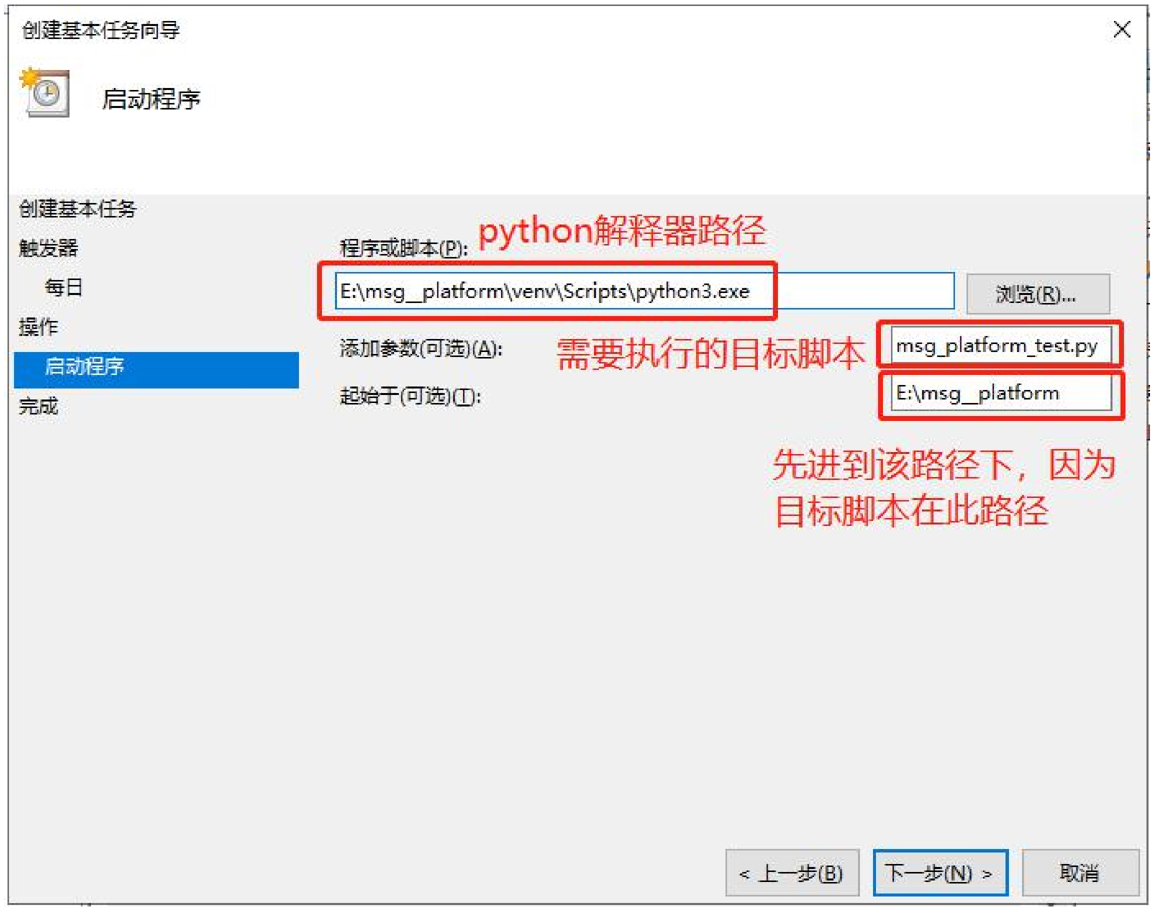


创建基本任务向导

操作

希望该任务执行什么操作?

(●) 启动程序(T)
( ) 发送电子邮件(已弃用)(S)
( ) 显示消息(已弃用)(M)



创建基本任务向导

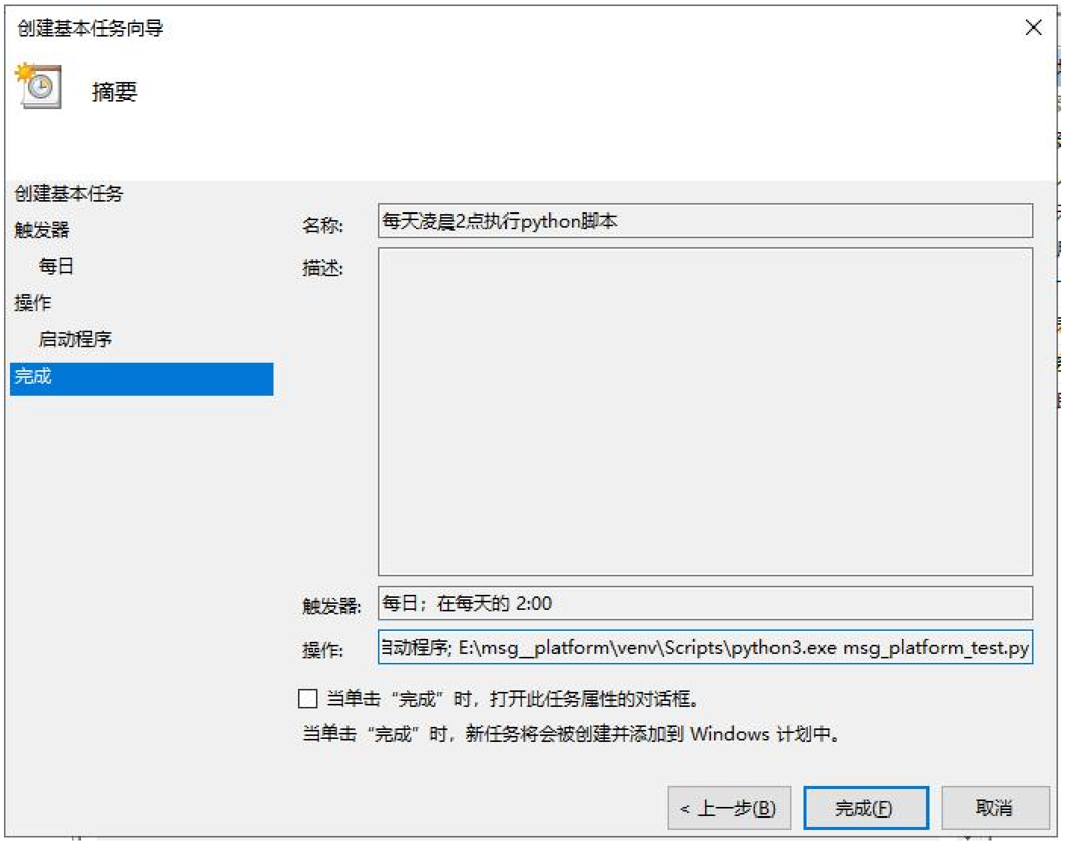
启动程序

创建基本任务
触发器
每日
操作
启动程序
完成

程序或脚本(P): E:\msg__platform\venv\Scripts\python3.exe

添加参数(可选)(A): msg_platform_test.py

起始于(可选)(T): E:\msg__platform



创建基本任务向导

摘要

创建基本任务
触发器
每日
操作
启动程序
完成

名称: 每天凌晨2点执行python脚本

描述:

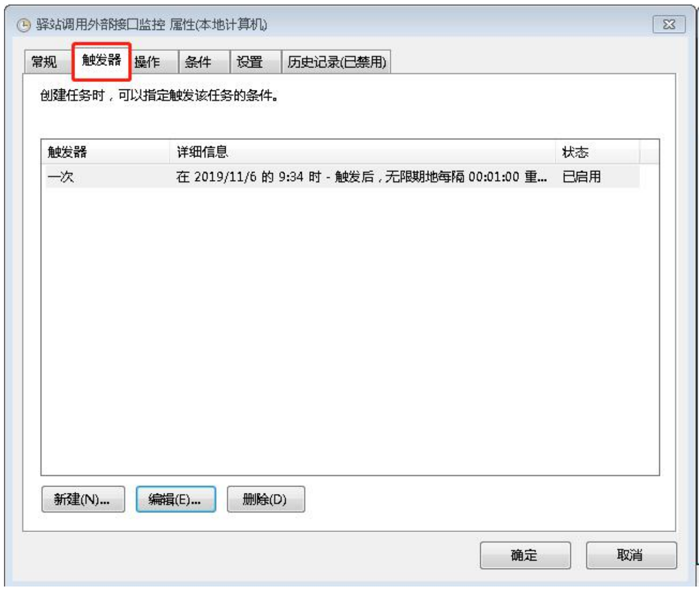
触发器: 每日；在每天的 2:00

操作: 启动程序; E:\msg_platform\venv\Scripts\python3.exe msg_platform_test.py

☐ 当单击“完成”时，打开此任务属性的对话框。

当单击“完成”时，新任务将会被创建并添加到 Windows 计划中。

< 上一步(B)　完成(F)　取消



驿站调用外部接口监控 属性(本地计算机)

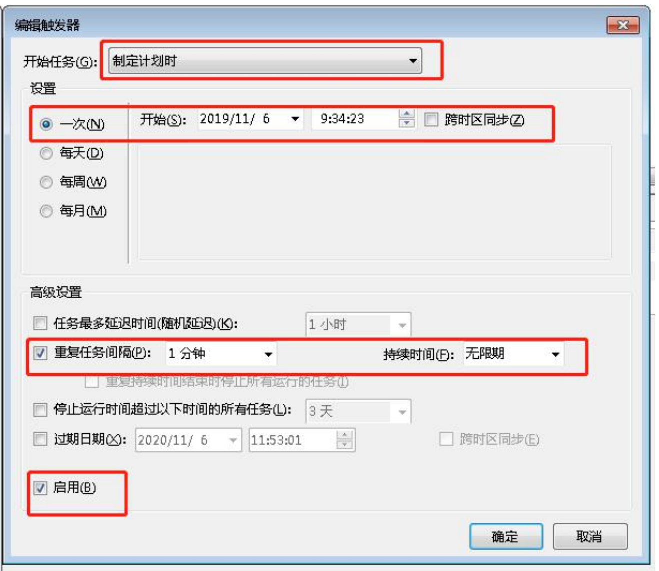
常规　触发器　操作　条件　设置　历史记录(已禁用)

创建任务时，可以指定触发该任务的条件。

触发器	详细信息	状态
一次	在 2019/11/6 的 9:34 时 - 触发后，无限期地每隔 00:01:00 重...	已启用

新建(N)...　编辑(E)...　删除(D)

确定　取消



编辑触发器

开始任务(G): 制定计划时

设置

[✓ 一次(N)]　每天(D)　每周(W)　每月(M)

开始(S): 2019/11/ 6　9:34:23　☐ 跨时区同步(Z)

高级设置

☐ 任务最多延迟时间(随机延迟)(K): 1 小时

☑ 重复任务间隔(P): 1 分钟　持续时间(F): 无限期

☐ 重复持续时间结束时停止所有运行的任务(I)

☐ 停止运行时间超过以下时间的所有任务(L): 3 天

☐ 过期日期(X): 2020/11/ 6　11:53:01　☐ 跨时区同步(E)

☑ 启用(B)

确定　取消

3.6 OS/linux 系统定时任务

crontab

- -e　　(编辑工作表)

- -l (列出工作表里的命令)
- -r (删除工作作)

crontab 的命令构成为 时间+动作，其时间有分、时、日、月、周五种，操作符有

1. * 取值范围内的所有数字
2. / 每过多少个数字
3. - 从 X 到 Z
4. ， 散列数字

实例 1：每 1 分钟执行一次 myCommand

```
* * * * * myCommand
```

实例 2：每小时的第 3 和第 15 分钟执行

```
3,15 * * * * myCommand
```

实例 3：在上午 8 点到 11 点的第 3 和第 15 分钟执行

```
3,15 8-11 * * * myCommand
```

实例 4：每隔两天的上午 8 点到 11 点的第 3 和第 15 分钟执行

```
3,15 8-11 */2    *    * myCommand
```

实例 5：每周一上午 8 点到 11 点的第 3 和第 15 分钟执行

```
3,15 8-11 * * 1 myCommand
```

实例 6：每晚的 21:30 重启 smb

```
30 21 * * * /etc/init.d/smb restart
```

实例 7：每月 1、10、22 日的 4 : 45 重启 smb

```
45 4 1,10,22 * * /etc/init.d/smb restart
```

实例 8：每周六、周日的 1 : 10 重启 smb

```
10 1 * * 6,0 /etc/init.d/smb restart
```

实例 9：每天 18 : 00 至 23 : 00 之间每隔 30 分钟重启 smb

```
0,30 18-23 * * * /etc/init.d/smb restart
```

实例 10：每星期六的晚上 11 : 00 pm 重启 smb

0 23 * * 6 /etc/init.d/smb restart

实例 11：每一小时重启 smb

* */1 * * * /etc/init.d/smb restart

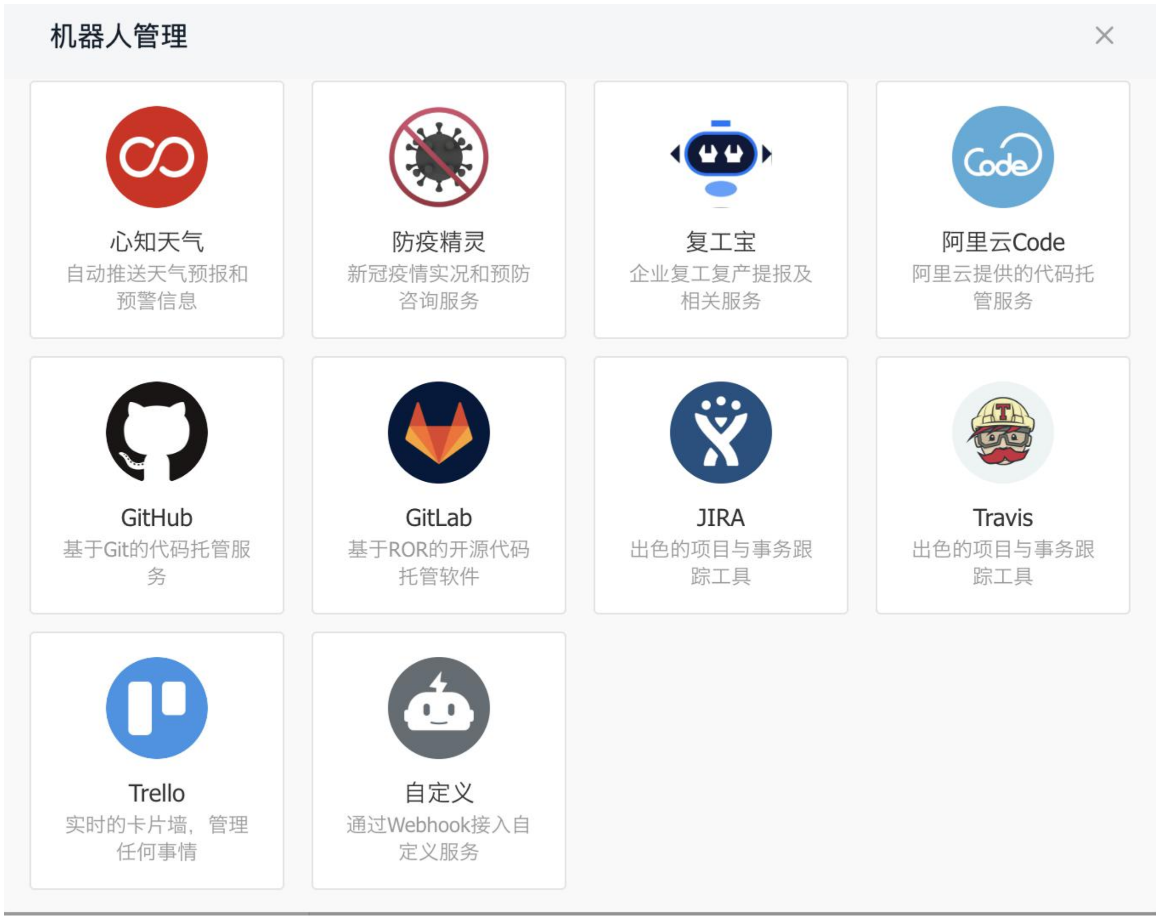
实例 12：晚上 11 点到早上 7 点之间，每隔一小时重启 smb

* 23-7/1 * * * /etc/init.d/smb restart

第四章：《机器人助理办公》系统自动化处理

4.1 Python 钉钉发送消息

创建钉钉机器人



```
# pip install dingtalkchatbot

from dingtalkchatbot.chatbot import DingtalkChatbot

url =
'https://oapi.dingtalk.com/robot/send?access_token=e472eb30c2229395d87c5f6a4ee0
228a8a860dc39f7392ed71f71e13bb6b1939'

boot =
DingtalkChatbot(url,secret='SECeb3e31cb12566e3e9d934a618adc39cbeba94389274932
81babe89b720640f5c')

boot.send_text(msg='这个是数据 sss')
```

5.2 Python 钉钉发送图片

```
    def send_img():
        from dingtalkchatbot.chatbot import DingtalkChatbot
        url = 'https://oapi.dingtalk.com/robot/send?access_token='
        boot = DingtalkChatbot(url,secret='')
        boot.send_image(pic_url='https://www.itbaizhan.cn/public/new/index/images/tuiimg6.png')


    def send_link():
        from dingtalkchatbot.chatbot import DingtalkChatbot
        url = 'https://oapi.dingtalk.com/robot/send?access_token='
        boot = DingtalkChatbot(url,secret='')
        # Link 消息
        boot.send_link(title='万万没想到，我们已成大神..', text='故事是这样子的...',
message_url='http://www.itbaizhan.cn',
pic_url="https://www.itbaizhan.cn/public/new/index/images/tuiimg6.png")
    def send_md():
        from dingtalkchatbot.chatbot import DingtalkChatbot
        url = 'https://oapi.dingtalk.com/robot/send?access_token='
        boot = DingtalkChatbot(url,secret='')
```

```
            # Link 消息
            boot.send_markdown(title='氧气文字', text='#### 北京天气\n'
                                      '> 9 度，西北风 1 级，空气良 89，相对温度 73%\n\n'
                                      '> ![美景](http://www.sinaimg.cn/dy/slidenews/5_img/2013_28/453_28488_469248.jpg)\n'
                                      '> ###### 10 点 20 分发布 [天气](http://www.itbaizhan.cn) \n',
                                      is_at_all=True)

        def send_card():
            from dingtalkchatbot.chatbot import DingtalkChatbot
            from dingtalkchatbot.chatbot import CardItem
            url = 'https://oapi.dingtalk.com/robot/send?access_token='
            boot = DingtalkChatbot(url,secret='')
            # Link 消息
            # FeedCard 消息类型
            'http://pic.netbian.com/uploads/allimg/190824/212516-1566653116f355.jpg'
            card1 = CardItem(title=" 氧 气 美 女 ", url="http://www.itbaizhan.cn", pic_url='http://pic.netbian.com/uploads/allimg/190824/212516-1566653116f355.jpg')
            card2 = CardItem(title=" 氧 眼 美 女 ", url="http://www.itbaizhan.cn", pic_url='http://pic.netbian.com/uploads/allimg/201112/000443-16051106836aa6.jpg')
            card3 = CardItem(title=" 氧 神 美 女 ", url="http://www.itbaizhan.cn", pic_url='http://pic.netbian.com/uploads/allimg/190824/205524-15666513248366.jpg')
            cards = [card1, card2, card3]

            boot.send_feed_card(cards)
        def send_card2():
            from dingtalkchatbot.chatbot import DingtalkChatbot
            from dingtalkchatbot.chatbot import CardItem,ActionCard
            url = 'https://oapi.dingtalk.com/robot/send?access_token='
            boot = DingtalkChatbot(url,secret='SECeb3')
            # Link 消息
            # ActionCard 整体跳转消息类型
            btns1 = [CardItem(title="查看详情", url="https://www.itbaizhan.cn/")]
            actioncard1 = ActionCard(title='万万没想到，竟然...',
                                     text='![选择](http://pic.netbian.com/uploads/allimg/201112/000443-16051106836aa6.jpg) \n### 故事是这样子的...',
                                     btns=btns1,
                                     btn_orientation=1,
                                     hide_avatar=1)
            boot.send_action_card(actioncard1)
        def send_card3():
            from dingtalkchatbot.chatbot import DingtalkChatbot
            from dingtalkchatbot.chatbot import CardItem,ActionCard
            url = 'https://oapi.dingtalk.com/robot/send?access_token=e'
            boot = DingtalkChatbot(url,secret='')



        def send_card3():
            from dingtalkchatbot.chatbot import DingtalkChatbot
            from dingtalkchatbot.chatbot import CardItem,ActionCard
            url = 'https://oapi.dingtalk.com/robot/send?access_token='
            boot = DingtalkChatbot(url,secret='')
            # Link 消息
            # ActionCard 独立跳转消息类型（双选项）
            btns3 = [CardItem(title="支持", url="https://www.itbaizhan.cn/"), CardItem(title="中立", url="https://www.itbaizhan.cn/"), CardItem(title="反对", url="https://www.itbaizhan.cn/")]
            actioncard3  =  ActionCard(title=' 万 万 没 想 到 ， 竟 然 ...',text='![ 选择](http://pic.netbian.com/uploads/allimg/190824/212516-1566653116f355.jpg) \n### 故事是这样子的...',
                                   btns=btns3,
                                   btn_orientation=1,
                                   hide_avatar=1)
            boot.send_action_card(actioncard3)
```

4.2 Python 操作压缩文件

```
    import zipfile
    import os


    def zip_ya():
        with zipfile.ZipFile('./yasuo/zip01.zip', 'w') as z:
            z.write('23_zmail.py')
            z.write('22_email 附件.py')


    def zip_jieya():
        with zipfile.ZipFile('./yasuo/a.zip', 'r') as z:
            print(z.namelist())       # 查看压缩包中的文件列表
            # z.extract('23_zmail.py','./yasuo')        # 解压，可选择解压某个文件
            # z.extractall('./yasuo/23_zmail.py')        # 解压全部


    def compress_file(zipfilename, dirname):          # zipfilename 是压缩包名字，dirname 是要打包的目录
        if os.path.isfile(dirname):
            with zipfile.ZipFile(zipfilename, 'w') as z:
                z.write(dirname)
        else:
            with zipfile.ZipFile(zipfilename, 'w') as z:
                for root, dirs, files in os.walk(dirname):
                    '''
                    root 所指的是当前正在遍历的这个文件夹的本身的地址
                    dirs 是一个 list ，内容是该文件夹中所有的目录的名字(不包括子目录)
                    files 同样是 list , 内容是该文件夹中所有的文件(不包括子目录)
                    '''
                    for single_file in files:
                        if single_file != zipfilename:
                            filepath = os.path.join(root, single_file)
                            z.write(filepath)

    def addfile(zipfilename, dirname):
        if os.path.isfile(dirname):
            with zipfile.ZipFile(zipfilename, 'a') as z:
                z.write(dirname)
        else:
            with zipfile.ZipFile(zipfilename, 'a') as z:
                for root, dirs, files in os.walk(dirname):
                    for single_file in files:
                        if single_file != zipfilename:
                            filepath = os.path.join(root, single_file)
                            z.write(filepath)
    import tarfile
    def tar_ya():
        with tarfile.open('./yasuo/b.tar', 'w') as tar:
```

```
        tar.add('23_zmail.py', arcname='23_zmail.py')
        tar.add('22_email 附件.py', arcname='22_email 附件.py')
def tar_jieya():
    with tarfile.open('./yasuo/a.tar', 'r') as tar:
        print(tar.getmembers())      # 查看压缩包内文件成员
        # tar.extract('test.txt')  # 可选择解压某个文件
        # tar.extractall('./yasuo')  # 可设置解压路径
        # tar.extractall()  # 解压全部

def compress_file(tarfilename, dirname):      # tarfilename 是压缩包名字，dirname 是要打包
的目录
    if os.path.isfile(dirname):
        with tarfile.open(tarfilename, 'w') as tar:
            tar.add(dirname)
    else:
        with tarfile.open(tarfilename, 'w') as tar:
            for root, dirs, files in os.walk(dirname):
                for single_file in files:
                    # if single_file != tarfilename:
                    filepath = os.path.join(root, single_file)
                    tar.add(filepath)

def addfile(tarfilename, dirname):      # tarfilename 是压缩包名字，dirname 是要打包的目录
    if os.path.isfile(dirname):
        with tarfile.open(tarfilename, 'a') as tar:
            tar.add(dirname)
    else:
```

```
                with tarfile.open(tarfilename, 'a') as tar:
                    for root, dirs, files in os.walk(dirname):
                        for single_file in files:
                            # if single_file != tarfilename:
                            filepath = os.path.join(root, single_file)
                            tar.add(filepath)
```

4.3 Python 暴力破解压缩密码

```
def passwd(path):
    # with    as target:
    type = os.path.splitext(path)[-1][1:]
    if type == "zip":
            with zipfile.ZipFile(path,'r') as z:
                for l in z.infolist():
                    # print(l.flag_bits)
                    is_encrypted = 1
                    if is_encrypted:
                        for i in range(9999):
                            try:
                                z.extractall('./yasuo',pwd=str(i).encode('utf-8'))
                                print(f'密码是：{i}')
                                break
                            except Exception as e:
                                pass
                    else:
                        z.extractall('./yasuo')
                    print('解压成功！ ')


def create_mi():
    import itertools as its
    words = "abc"
    r =its.product(words,repeat=2)
    for i in r:
        print(''.join(i))
```