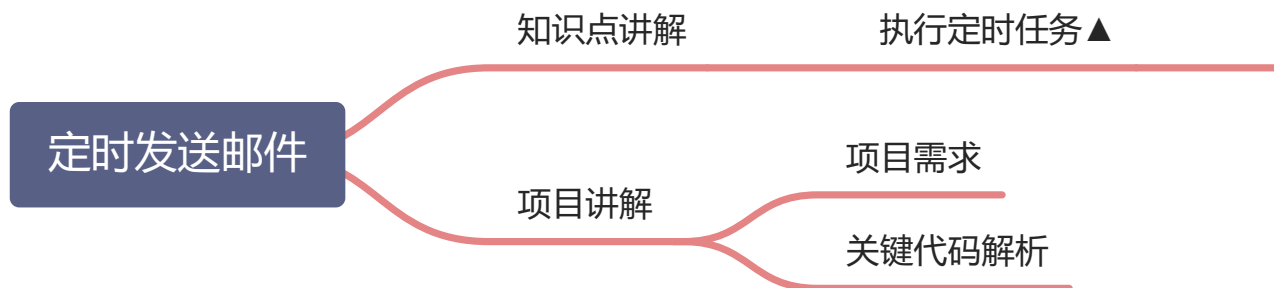


第10课 定时发送邮件

一、课程结构导图



注：▲为重点知识点。

二、知识点讲解

2.1 执行定时任务

用法：

通过schedule库我们可以做到按照设定好的时间间隔来运行一个函数，首先使用pip install schedule安装schedule库。

使用代码schedule.every().xx.at('HH:MM:SS').do(job)可以来设定一个需要定时执行的任务。

其中every()方法的参数是可选的，它是否带参数取决于后面xx属性，xx属性如果是单数就不带参数，如果是复数就带参数。

xx表示时间频率，有单数的形式的second,minute,hour,day,week，以及它们的复数形式。还有表示日期的monday,tuesday,wednesday,thursday,friday,saturday,sunday。

at()方法是一个可选方法，表示需要一个确定的时刻。它的参数是一个时间点HH:MM:SS。

do()方法的参数表示需要执行的函数名以及它的参数。比如do(x,y)，x表示要执行x()函数，y是x()函数的参数。

示例：

```
1 import schedule
2 import time
3
4 def job():
5     print("I'm working...")
6
7 def greet(name):
8     print('Hello', name)
9
10 schedule.every(3).seconds.do(job) # 每隔三秒执行一次
11 schedule.every().hour.do(job) # 每隔一个小时执行一次
12 schedule.every().day.at("10:30").do(job) # 每天早上10点半执行一次
13 schedule.every().monday.do(job) # 每周一执行一次
14 schedule.every().wednesday.at("13:15").do(job) # 每周三执行一次
15 schedule.every().minute.at(":17").do(job) # 每分钟第17秒执行一次
16
17 schedule.every(2).seconds.do(greet, name='Alice')
18 schedule.every(4).seconds.do(greet, name='Bob')
19
20 while True:
21     schedule.run_pending() #检测并运行设定好的所有定时任务
22     time.sleep(1)
```

代码解析：代码运行到第20行的schedule.run_pending()开始检测并运行上面设定好的所有定时任务。我们需要定时任务不断被检测并运行，因此要求schedule.run_pending()写在死循环里。

三、项目讲解

3.1 项目需求

设定一个定时程序，使其可以每天自动爬取中国气象网

<http://www.weather.com.cn/weather/101280601.shtml>

[<http://www.weather.com.cn/weather/101280601.shtml>](http://www.weather.com.cn/weather/101280601.shtml) 中的温度和天气信息并通过邮件发送给别人。

3.2 关键代码解析

完整代码：

```
1 import requests
2 import smtplib
3 import schedule
4 import time
5 from bs4 import BeautifulSoup
6 from email.mime.text import MIMEText
7 from email.header import Header
8
9 account = input('请输入你的邮箱: ')
10 password = input('请输入你的密码: ')
11 receiver = input('请输入收件人的邮箱: ')
12
13 def weather_spider():
14     headers={'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
15     AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
16     url='http://www.weather.com.cn/weather/101280601.shtml'
17     res=requests.get(url,headers=headers)
18     res.encoding='utf-8'
19     soup=BeautifulSoup(res.text,'html.parser')
20     tem1= soup.find(class_='tem')
21     weather1= soup.find(class_='wea')
22     tem=tem1.text
23     weather=weather1.text
24     return tem,weather
25
26 def send_email(tem,weather):
27     mailhost='smtp.qq.com'
28     qqmail = smtplib.SMTP()
29     qqmail.connect(mailhost,25)
30     qqmail.login(account,password)
31     content= tem+weather
32     message = MIMEText(content, 'plain', 'utf-8')
```

```

32     subject = '今日天气预报'
33     message['Subject'] = Header(subject, 'utf-8')
34     try:
35         qqmail.sendmail(account, receiver, message.as_string())
36         print ('邮件发送成功')
37     except:
38         print ('邮件发送失败')
39     qqmail.quit()
40
41 def job():
42     print('开始一次任务')
43     tem,weather = weather_spider()
44     send_email(tem,weather)
45     print('任务完成')
46
47 schedule.every().day.at("07:30").do(job)
48 while True:
49     schedule.run_pending()
50     time.sleep(1)

```

代码解析：

1. 通过检查模式发现存放温度信息的标签有class='tem'属性，存放天气信息的标签有class='wea'属性。
2. 爬取到温度和天气两个数据后第23行返回这两个数据。稍微来复习下，Python语言的多值返回是元组的形式。同时第43行代码是按照元组中数据的存放顺序将数据分别赋值给tem和weather变量。
3. 第48行代码开始通过死循环每隔一秒检查并运行定时任务。