

# 第 10 关、定时与邮件

## 1、项目：自动爬取每日的天气

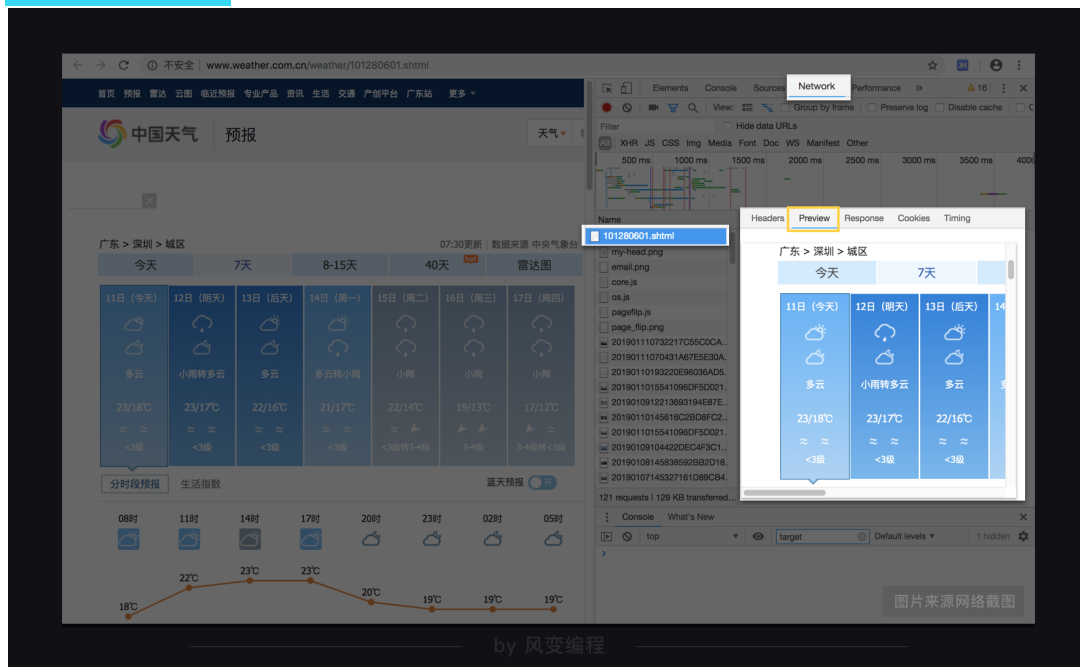
### 1-1、确定目标

- (1) 目标网站：<http://www.weather.com.cn/weather/101280601.shtml>；
- (2) 项目目标：自动爬取每日的天气，并定时把天气数据和穿衣提示发送到你的邮箱；
- (3) 项目划分：【爬虫】+【邮件】+【定时】。

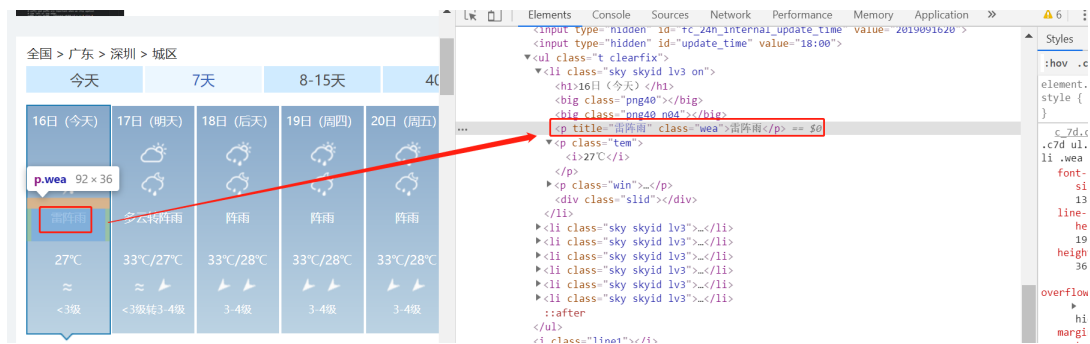
### 1-2、过程分析

#### 1-2-1、爬取数据定位

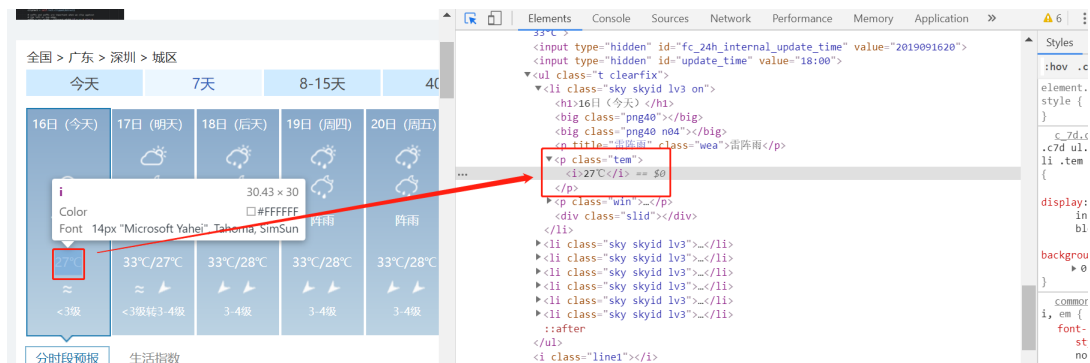
- (1) 点击“右键”——“检查”——“Network”，刷新页面，点击查看第 0 个请求：[101280601.shtml](http://101280601.shtml)；



- (2) 天气情况对应的标签为 `class = 'wea'`；



(3) 温度对应的标签为 class = 'tem';



```

1 import requests
2 from bs4 import BeautifulSoup
3 headers={'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
4 url='http://www.weather.com.cn/weather/101280601.shtml'
5 res=requests.get(url,headers=headers)
6 res.encoding='utf-8'
7 bsdata=BeautifulSoup(res.text,'html.parser')
8 #使用bs模块解析获取到的数据
9 data1= bsdata.find(class_='tem')
10 #使用find()取出天气的温度数据
11 data2= bsdata.find(class_='wea')
12 #使用find()取出天气的文字描述
13 print(data1.text)
14 #取出变量data1中的字符串内容,并打印
15 print(data2.text)
16 #取出变量data2中的字符串内容,并打印

```

## 1-2-2、发送邮件

参考基础语法的邮件发送: <https://shimo.im/docs/nPtTnzU1dasU9PEI/> 《第17关、邮件还能这么发》

```

1 import smtplib
2 from email.mime.text import MIMEText
3 from email.header import Header
4 #引入smtplib、MIMEText和Header
5
6 mailhost='smtp.qq.com'
7 #把qq邮箱的服务器地址赋值到变量mailhost上,地址应为字符串格式

```

```

8 qqmail = smtplib.SMTP()
9 #实例化一个smtplib模块里的SMTP类的对象，这样就可以调用SMTP对象的方法和属性了
10 qqmail.connect(mailhost,25)
11 #连接服务器，第一个参数是服务器地址，第二个参数是SMTP端口号。
12 #以上，皆为连接服务器。
13
14 account = input('请输入你的邮箱: ')
15 #获取邮箱账号，为字符串格式
16 password = input('请输入你的密码: ')
17 #获取邮箱密码，为字符串格式
18 qqmail.login(account,password)
19 #登录邮箱，第一个参数为邮箱账号，第二个参数为邮箱密码
20 #以上，皆为登录邮箱。
21
22 receiver=input('请输入收件人的邮箱: ')
23 #获取收件人的邮箱。
24
25 content=input('请输入邮件正文: ')
26 #输入你的邮件正文，为字符串格式
27 message = MIMEText(content, 'plain', 'utf-8')
28 #实例化一个MIMEText邮件对象，该对象需要写进三个参数，分别是邮件正文，文本格式和编码
29 subject = input('请输入你的邮件主题: ')
30 #输入你的邮件主题，为字符串格式
31 message['Subject'] = Header(subject, 'utf-8')
32 #在等号的右边是实例化了一个Header邮件头对象，该对象需要写入两个参数，分别是邮件主题和编码，然后赋值给等号左边的变量message['Subject']。
33 #以上，为填写主题和正文。
34
35 try:
36     qqmail.sendmail(account, receiver, message.as_string())
37     print ('邮件发送成功')
38 except:
39     print ('邮件发送失败')
40 qqmail.quit()
41 #以上为发送邮件和退出邮箱。

```

### 1-2-3、定时 (schedule)

schedule 比起 time 和 datetime 可以直接解决定时功能，代码比较简单

## Usage

```
$ pip install schedule
```

```
import schedule
import time

def job():
    print("I'm working...")

schedule.every(10).minutes.do(job)
schedule.every().hour.do(job)
schedule.every().day.at("10:30").do(job)
schedule.every().monday.do(job)
schedule.every().wednesday.at("13:15").do(job)

while True:
    schedule.run_pending()
    time.sleep(1)
```

by 风变编程

```
1 import schedule
2 import time
3 #引入schedule和time
4
5 def job():
6     print("I'm working...")
7 #定义一个叫job的函数，函数的功能是打印'I'm working...'
8
9 schedule.every(10).minutes.do(job)      #部署每10分钟执行一次job()函数的任务
10 schedule.every().hour.do(job)          #部署每x小时执行一次job()函数的任务
11 schedule.every().day.at("10:30").do(job) #部署在每天的10:30执行job()函数的任务
12
13 schedule.every().monday.do(job)         #部署每个星期一执行job()函数的任务
14 schedule.every().wednesday.at("13:15").do(job) #部署每周三的13:15执行函数的任务
15
16 while True:
17     schedule.run_pending()
18     time.sleep(1)
19 #15-17都是检查部署的情况，如果任务准备就绪，就开始执行任务。
```

- 第 1 行和第 2 行，是引入 schedule 和 time；
- 第 5 行和第 6 行，是定义了一个叫 job() 的函数，调用这个函数时，函数会打印 I'm working...；
- 第 9 行 – 13 行都是相关的时间设置，你可以根据自己的需要来确定；
- 第 15 – 17 行是一个while循环，是去检查上面的任务部署情况，如果任务已经准备就绪，就去启动执行。其中，第 15 行的 time.sleep(1) 是让程序按秒来检查，如果检查太快，会浪费计算机的资源。

## 1-3、代码组装

```
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     global account,password,receiver
28     mailhost='smtp.qq.com'
29     qqmail = smtplib.SMTP()
30     qqmail.connect(mailhost,25)
31     qqmail.login(account,password)
32     content= tem+weather
33     message = MIMEText(content, 'plain', 'utf-8')
34     subject = '今日天气预报'
35     message['Subject'] = Header(subject, 'utf-8')
36     try:
37         qqmail.sendmail(account, receiver, message.as_string())
38         print ('邮件发送成功')
39     except:
40         print ('邮件发送失败')
41     qqmail.quit()
42
43 def job():
44     print('开始一次任务')
45     tem,weather = weather_spider()
46     send_email(tem,weather)
47     print('任务完成')
48
49 schedule.every().day.at("07:30").do(job)
50 while True:
51     schedule.run_pending()
52     time.sleep(1)

```

- 第 1 – 7 行是把所有引入模块都放到程序的顶部；

- 第 9 – 11 行也是把邮件数据放到函数的外面；
- 第 13 – 23 行是定义了一个 `weather_spider` 的函数，用于获取网页对应的天气情况和温度数据；
- 第 25 – 40 行定义了一个 `send_email` 的函数，用于将获取的天气情况和温度数据发送到对应的邮箱中；
- 第 42 行开始，定义一个函数叫 `job()`；
- 第 43 行是打印‘开始一次任务’，为了记录和显示任务的开始；
- 第 44 行，是调用爬虫函数 `weather_spider()`，然后把这个函数内部 `return` 的两个变量 `tem`、`weather` 赋值给 `job()` 函数里面的变量 `tem`，`weather`；
- 第 45 行是调用函数 `send_email()`，并且把参数传入；
- 第 46 行打印‘任务完成’，表示这部分程序运行正常；
- 第 48 – 51 行都是定时功能我们见过的函数，我们设定的是每天早上七点半把天气信息传递给收件人。