太湖创意职业技术学院

**实 验（实 训）报 告**

**实验（实训）名称 python大数据**

**所属课程名称 python大数据**

**班 级**  计算机应用202

**学 号**  12042034

**姓 名**  马云骥

**指导教师**  李成渊

**项目名称：**爬取天气预报数据  **日期：**2022-4-7

|  |
| --- |
| **一、实验（实训）概述：** |
| **【目的及要求】**  1.掌握Python的基本编程语法。  2.利用Python编写一段小程序。  **【实施环境】（使用的材料、设备、软件）**  **设备：**计算机 、Pycharm |
| **二、实验（实训）内容：** |
| **【实验内容】**  1．在中国天气网(http://www.weather.com.cn)中输入一个城市的名称，例如输入深圳，那么会转到地址http://www.weather.com.cn/weather1d/101280601.shtml的网页显示深圳的天气预报，其中101280601是深圳的代码，每个城市或者地区都有一个代码。  2．在任务1基础之上，试获取北京、上海、广州、深圳、你的家乡、无锡的城市代码，爬取这些城市的天气预报数据，存储到sqllite数据库weathers.db，并完成界面输出。  **【实验（实训）过程】（步骤、记录、数据、程序等）**  **任务 1**  **分析 HTML 代码**  用 Chrome 浏览器浏览网站，鼠标指向 7 天天气预报的今天位置，点击右键弹出菜单，选择“检查”就可以打开这个位置对应的 HTML 代码，如图所示。  截屏2022-04-07 09.11.58  选择 <ul class="t clearfix"> 元素，点击右键弹出菜单选择 "Edit as HTML"，就可以进入编辑状态，复制整个 HTML，结果如下：  <ul class="t clearfix">  <li class="sky skyid lv1 on">  <h1>7日（今天）</h1>  <big class="png40 d01"></big>  <big class="png40 n01"></big>  <p title="多云" class="wea">多云</p>  <p class="tem">  <span>30</span>/<i>18℃</i>  </p>  <p class="win">  <em>  <span title="无持续风向" class="NNW"></span>  <span title="无持续风向" class="NNW"></span>  </em>  <i>&lt;3级</i>  </p>  <div class="slid"></div>  </li>  <li class="sky skyid lv1">  <h1>8日（明天）</h1>  <big class="png40 d01"></big>  <big class="png40 n01"></big>  <p title="多云" class="wea">多云</p>  <p class="tem">  <span>29</span>/<i>18℃</i>  </p>  <p class="win">  <em>  <span title="无持续风向" class="NNW"></span>  <span title="无持续风向" class="NNW"></span>  </em>  <i>&lt;3级</i>  </p>  <div class="slid"></div>  </li>  <li class="sky skyid lv1">  <h1>9日（后天）</h1>  <big class="png40 d01"></big>  <big class="png40 n01"></big>  <p title="多云" class="wea">多云</p>  <p class="tem">  <span>30</span>/<i>18℃</i>  </p>  <p class="win">  <em>  <span title="无持续风向" class="NNW"></span>  <span title="无持续风向" class="NNW"></span>  </em>  <i>&lt;3级</i>  </p>  <div class="slid"></div>  </li>  <li class="sky skyid lv1">  <h1>10日（周日）</h1>  <big class="png40 d00"></big>  <big class="png40 n00"></big>  <p title="晴" class="wea">晴</p>  <p class="tem">  <span>28</span>/<i>20℃</i>  </p>  <p class="win">  <em>  <span title="无持续风向" class="NNW"></span>  <span title="无持续风向" class="NNW"></span>  </em>  <i>&lt;3级</i>  </p>  <div class="slid"></div>  </li>  <li class="sky skyid lv1">  <h1>11日（周一）</h1>  <big class="png40 d00"></big>  <big class="png40 n01"></big>  <p title="晴转多云" class="wea">晴转多云</p>  <p class="tem">  <span>28</span>/<i>21℃</i>  </p>  <p class="win">  <em>  <span title="无持续风向" class="NNW"></span>  <span title="无持续风向" class="NNW"></span>  </em>  <i>&lt;3级</i>  </p>  <div class="slid"></div>  </li>  <li class="sky skyid lv1">  <h1>12日（周二）</h1>  <big class="png40 d01"></big>  <big class="png40 n01"></big>  <p title="多云" class="wea">多云</p>  <p class="tem">  <span>29</span>/<i>21℃</i>  </p>  <p class="win">  <em>  <span title="无持续风向" class="NNW"></span>  <span title="无持续风向" class="NNW"></span>  </em>  <i>&lt;3级</i>  </p>  <div class="slid"></div>  </li>  <li class="sky skyid lv2">  <h1>13日（周三）</h1>  <big class="png40 d07"></big>  <big class="png40 n01"></big>  <p title="小雨转多云" class="wea">小雨转多云</p>  <p class="tem">  <span>27</span>/<i>19℃</i>  </p>  <p class="win">  <em>  <span title="无持续风向" class="NNW"></span>  <span title="东北风" class="NE"></span>  </em>  <i>&lt;3级转3-4级</i>  </p>  <div class="slid"></div>  </li> </ul>  分析这段代码容易发现 7 天的天气预报实际上在一个 <ul class="t clearfix"> 元素之中，每天是一个 <li> 元素，每天的 <li> 结构是一样的，因此可以通过 BeautifulSoup 的元素查找方法得到各个元素的值。  **初步爬取天气预报数据**  通过分析 HTML 代码，我们可以编写爬取的程序爬取深圳 7 天的天气预报数据：  from bs4 import BeautifulSoup from bs4 import UnicodeDammit import urllib.request ​ url = "http://www.weather.com.cn/weather/101280601.shtml" try:     headers = {         "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 Safari/605.1.15"}     req = urllib.request.Request(url, headers=headers)     data = urllib.request.urlopen(req)     data = data.read()     dammit = UnicodeDammit(data, ["utf-8", "gbk"])     data = dammit.unicode\_markup     soup = BeautifulSoup(data, "lxml")     lis = soup.select("ul[class='t clearfix'] li")     for li in lis:         try:             date = li.select('h1')[0].text             weather = li.select('p[class="wea"]')[0].text             temp = li.select('p[class="tem"] span')[0].text + "/" + li.select('p[class="tem"] i')[0].text             print(date, weather, temp)         except Exception as err:             print(err) except Exception as err:     print(err)  **任务 2**  **爬取与存储天气预报数据**  我们可以获取北京、上海、广州、深圳、徐州、无锡等城市的代码，爬取这些城市的天气预报数据，并存储到 sqlite 数据库 weathers.db 中，存储的数据表 weathers 是：  create table weathers (wCity varchar(16),wDate varchar(16),wWeather varchar(64),wTemp varchar(32),constraint pk\_weather primary key (wCity,wDate))  编写程序依次爬取各个城市的天气预报数据存储在数据库中，程序如下：  from bs4 import BeautifulSoup from bs4 import UnicodeDammit import urllib.request import sqlite3 ​ ​ class WeatherDB:     def openDB(self):         self.con = sqlite3.connect("weathers.db")         self.cursor = self.con.cursor()         try:             self.cursor.execute(                 "create table weathers (wCity varchar(16),wDate varchar(16),wWeather varchar(64),wTemp varchar(32),constraint pk\_weather primary key (wCity,wDate))")         except:             self.cursor.execute("delete from weathers") ​     def closeDB(self):         self.con.commit()         self.con.close() ​     def insert(self, city, date, weather, temp):         try:             self.cursor.execute("insert into weathers (wCity,wDate,wWeather,wTemp) values (?,?,?,?)",                                (city, date, weather, temp))         except Exception as err:             print(err) ​     def show(self):         self.cursor.execute("select \* from weathers")         rows = self.cursor.fetchall()         print("%-16s%-16s%-32s%-16s" % ("city", "date", "weather", "temp"))         for row in rows:             print("%-16s%-16s%-32s%-16s" % (row[0], row[1], row[2], row[3])) ​ ​ class WeatherForecast:     def \_\_init\_\_(self):         self.headers = {             "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 Safari/605.1.15"}         self.cityCode = {"北京": "101010100", "上海": "101020100", "广州": "101280101", "深圳": "101280601", "徐州": "101190801", "无锡": "101190201"} ​     def forecastCity(self, city):         if city not in self.cityCode.keys():             print(city + " code cannot be found")             return         url = "http://www.weather.com.cn/weather/" + self.cityCode[city] + ".shtml"         try:             req = urllib.request.Request(url, headers=self.headers)             data = urllib.request.urlopen(req)             data = data.read()             dammit = UnicodeDammit(data, ["utf-8", "gbk"])             data = dammit.unicode\_markup             soup = BeautifulSoup(data, "lxml")             lis = soup.select("ul[class='t clearfix'] li")             for li in lis:                 try:                     date = li.select('h1')[0].text                     weather = li.select('p[class="wea"]')[0].text                     temp = li.select('p[class="tem"] span')[0].text + "/" + li.select('p[class="tem"] i')[0].text                     print(city, date, weather, temp)                     self.db.insert(city, date, weather, temp)                 except Exception as err:                     print(err)         except Exception as err:             print(err) ​     def process(self, cities):         self.db = WeatherDB()         self.db.openDB()         for city in cities:             self.forecastCity(city)         # self.db.show()         self.db.closeDB() ​ ​ ws = WeatherForecast() ws.process(["北京", "上海", "广州", "深圳", "徐州", "无锡"]) print("completed")  **【结论】（结果、分析）**  **任务 1**  运行程序  截屏2022-04-07 09.47.09  程序爬取结果：  7日（今天） 多云 30/18℃ 8日（明天） 多云 29/18℃ 9日（后天） 多云 30/18℃ 10日（周日） 晴 28/20℃ 11日（周一） 晴转多云 28/21℃ 12日（周二） 多云 29/21℃ 13日（周三） 小雨转多云 27/19℃  由此可见爬取的数据与我们直接从网站看到的是一样的。  **任务 2**  运行程序  截屏2022-04-07 09.57.04  程序执行的效果如下：  北京 7日（今天） 晴 24/9℃ 北京 8日（明天） 晴 27/12℃ 北京 9日（后天） 晴 27/10℃ 北京 10日（周日） 多云 25/13℃ 北京 11日（周一） 多云转阴 26/10℃ 北京 12日（周二） 阴转多云 20/9℃ 北京 13日（周三） 多云转晴 20/10℃ 上海 7日（今天） 阴转多云 26/13℃ 上海 8日（明天） 阴 24/14℃ 上海 9日（后天） 多云转阴 26/16℃ 上海 10日（周日） 多云转阴 24/16℃ 上海 11日（周一） 阴 29/18℃ 上海 12日（周二） 阴 28/13℃ 上海 13日（周三） 小雨 17/10℃ 广州 7日（今天） 多云 30/18℃ 广州 8日（明天） 多云 30/17℃ 广州 9日（后天） 多云 31/18℃ 广州 10日（周日） 晴 31/18℃ 广州 11日（周一） 晴转多云 31/19℃ 广州 12日（周二） 多云转雷阵雨 30/20℃ 广州 13日（周三） 小雨转晴 27/18℃ 深圳 7日（今天） 多云 30/18℃ 深圳 8日（明天） 多云 29/18℃ 深圳 9日（后天） 多云 30/18℃ 深圳 10日（周日） 晴 28/20℃ 深圳 11日（周一） 晴转多云 28/21℃ 深圳 12日（周二） 多云 29/21℃ 深圳 13日（周三） 小雨转多云 27/19℃ 徐州 7日（今天） 多云 24/15℃ 徐州 8日（明天） 多云 29/18℃ 徐州 9日（后天） 多云 29/18℃ 徐州 10日（周日） 多云 30/20℃ 徐州 11日（周一） 多云 31/22℃ 徐州 12日（周二） 小雨 26/12℃ 徐州 13日（周三） 多云转晴 17/10℃ 无锡 7日（今天） 多云 27/13℃ 无锡 8日（明天） 多云 28/15℃ 无锡 9日（后天） 多云 28/15℃ 无锡 10日（周日） 多云 27/16℃ 无锡 11日（周一） 多云 30/18℃ 无锡 12日（周二） 多云转小雨 31/18℃ 无锡 13日（周三） 中雨转多云 18/11℃ completed  程序自动生成了存储天气预报数据的数据库 weathers.db 截屏2022-04-07 09.48.49 |
| **三、指导教师评语及成绩：** |
| **评语：**  该同学基本掌握Python的基本编程语法，能利用Python编写一段小程序，完成爬取天气预报数据的功能，达到了实训的要求。  **成绩：** A    **批阅日期：**2022-4-7 |