

先放结果

1 > >> | ☐ 显示全部 | 行数: 25

+ 选项

city	weather	wind	max	min
北京	多云	无持续风向<3级	0	-2
海淀	多云	西北风<3级	0	-2
朝阳	多云	北风<3级	0	-3
顺义	多云	北风<3级	0	-2
怀柔	多云	北风<3级	0	-4
通州	多云	北风<3级	0	-3
昌平	多云	西北风<3级	0	-2
延庆	多云	西风<3级	0	-5
丰台	多云	西北风<3级	0	-2
石景山	多云	西北风<3级	0	-2
大兴	多云	北风<3级	0	-3
房山	多云	北风<3级	0	-2
密云	多云	东北风<3级	0	-4
门头沟	多云	西北风<3级	0	-1
平谷	多云	北风<3级	0	-3
东城	多云	北风<3级	0	-3
西城	多云	西北风<3级	0	-2
天津	多云	无持续风向3-4级	0	-1
武清	多云	北风<3级	0	-4
宝坻	多云	北风3-4级	0	-5
东丽	多云	北风3-4级	0	-3
西青	多云	北风3-4级	0	-3
Console		北风<3级	0	-6

因为是昨天晚上发布的，所以昨天白天的气温就没有了，原网页是这样的

今天周一(12月3日)周二(12月4日)周三(12月5日)周四(12月6日)周五(12月7日)周六(12月8日)周日(12月9日)

省/直辖市	城市	周一(12月3日)白天			周一(12月3日)夜间		
		天气现象	风向风力	最高气温	天气现象	风向风力	最低气温
北京	北京	-	-	-	多云	无持续风向 <3级	-2
	海淀	-	-	-	多云	西北风 <3级	-2
	朝阳	-	-	-	多云	北风 <3级	-3
	顺义	-	-	-	多云	北风 <3级	-2
	怀柔	-	-	-	多云	北风 <3级	-4
	通州	-	-	-	多云	北风 <3级	-3
	昌平	-	-	-	多云	西北风 <3级	-2
	延庆	-	-	-	多云	西风 <3级	-5
	丰台	-	-	-	多云	西北风 <3级	-2
	石景山	-	-	-	多云	西北风 <3级	-2
	大兴	-	-	-	多云	北风 <3级	-3
	房山	-	-	-	多云	北风 <3级	-2
	密云	-	-	-	多云	东北风 <3级	-4
	门头沟	-	-	-	多云	西北风 <3级	-1
	平谷	-	-	-	多云	北风 <3级	-3
	东城	-	-	-	多云	北风 <3级	-3
	西城	-	-	-	多云	西北风 <3级	-2

一、建立数据库

观察我们要爬的数据，需要建立一个表单，表单里有 城市、天气、风向、最高气温、最低气温

我建的数据库如下

数据库名	用户名	密码	备份	备注
sql_Weather	sql_Weather	hellosandy	无备份 导入	sql_Weather

同步选中 同步所有 从服务器获取

我的表单名是 Weather

#	名字	类型	排序规则	属性	空	默认	额外	操作
1	city	text	utf8_general_ci		否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
2	weather	text	utf8_general_ci		否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
3	wind	text	utf8_general_ci		否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
4	max	int(11)			否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
5	min	int(11)			否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)

这里一定要改成改成 text 类型，否则默认是 int 会乱码

二、环境配置

在云服务器安装 pymysql 和 bs4

```
* Socket connection established *
Last login: Mon Dec  3 22:52:57 2018 from 119.28.7.195
[root@VM_0_13-centos ~]# pip install bs4
Collecting bs4
  Downloading https://pypi.doubanio.com/packages/10/ed/7e8b97591f6f456174139ec089c769f89a9/bs4-0.0.1.tar.gz (51kB)
    100% |#####| 92kB 5.9MB/s
Installing collected packages: bs4
  Running setup.py install for bs4 ... done
Successfully installed bs4-0.0.1
You are using pip version 9.0.1, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[root@VM_0_13-centos ~]# pip install pymysql
Collecting pymysql
  Downloading https://pypi.doubanio.com/packages/a7/7d/682c4a7da195a678047c8f1c51bb7682aae/pymysql-0.9.2.tar.gz (51kB)
    100% |#####| 51kB 1.4MB/s
Requirement already satisfied: cryptography in /usr/lib64/python2.7/site-packages (from pymysql)
Requirement already satisfied: idna>=2.1 in /usr/lib/python2.7/site-packages (from cryptography)
Requirement already satisfied: cffi>=1.7; platform_python_implementation != "PyPy" in /usr/lib/python2.7/site-packages (from cryptography)
Requirement already satisfied: enum34; python_version < "3" in /usr/lib/python2.7/site-packages (from cryptography)
Requirement already satisfied: six>=1.4.1 in /usr/lib/python2.7/site-packages (from cryptography)
Requirement already satisfied: asn1crypto>=0.21.0 in /usr/lib/python2.7/site-packages (from cryptography)
Requirement already satisfied: ipaddress; python_version < "3" in /usr/lib/python2.7/site-packages (from cryptography)
Requirement already satisfied: pyparsing in /usr/lib/python2.7/site-packages (from cryptography)
Installing collected packages: pymysql
Successfully installed pymysql-0.9.2
```

三、用 python 脚本写一个爬虫程序

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
import sys
reload(sys)
sys.setdefaultencoding("utf-8")
from bs4 import BeautifulSoup
import requests
import pymysql
```

用户名

```
conn = pymysql.connect(host='localhost', user='sql_Weather',
passwd='hellosandy', db='sql_Weather', port=3306,
charset='utf8')
```

密码

数据库名

```
cursor = conn.cursor()
sql = "TRUNCATE TABLE Weather"
try:
    cursor.execute(sql)
    conn.commit()
except:
    print
    "Error: unable to delete"
```

这里是删除表单的所有行
因为脚本每次都爬一遍网页，如
果不删除的话表单会越来越长
QAQ

```
def get_temperature(url):
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
        AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132
        Safari/537.36'} # 设置头文件信息
    response = requests.get(url, headers=headers).content # 提交
    requests.get 请求
    soup = BeautifulSoup(response, "html.parser") # 用
    BeautifulSoup 进行解析

    conmid = soup.find('div', class_='conMidtab')
    conmid2 = conmid.findAll('div', class_='conMidtab2')

    for info in conmid2:
        tr_list = info.find_all('tr')[2:] # 使用切片取到第三个 tr 标
        签
        for index, tr in enumerate(tr_list): # enumerate 可以返回元
        素的位置及内容
            td_list = tr.find_all('td')
            if index == 0:
```

```

        province_name = td_list[0].text.replace('\n', '')
# 取每个标签的 text 信息, 并使用 replace() 函数将换行符删除
        city_name = td_list[1].text.replace('\n', '')
        weather = td_list[5].text.replace('\n', '')
        wind = td_list[6].text.replace('\n', '')
        max = td_list[4].text.replace('\n', '')
        min = td_list[7].text.replace('\n', '')
        print(province_name)
    else:
        city_name = td_list[0].text.replace('\n', '')
        weather = td_list[4].text.replace('\n', '')
        wind = td_list[5].text.replace('\n', '')
        max = td_list[3].text.replace('\n', '')
        min = td_list[6].text.replace('\n', '')

        cursor.execute('insert into
Weather(city,weather,wind,max,min) values(%s,%s,%s,%s,%s)',
(city_name, weather, wind, max, min))

```

表单名

```

if __name__ == '__main__':
    urls = ['http://www.weather.com.cn/textFC/hb.shtml',
            'http://www.weather.com.cn/textFC/db.shtml',
            'http://www.weather.com.cn/textFC/hd.shtml',
            'http://www.weather.com.cn/textFC/hz.shtml',
            'http://www.weather.com.cn/textFC/hn.shtml',
            'http://www.weather.com.cn/textFC/xb.shtml',
            'http://www.weather.com.cn/textFC/xn.shtml']
    for url in urls:
        get_temperature(url)
    conn.commit()

```

代码

```

#!/usr/bin/python
# -*- coding: UTF-8 -*-
import sys
reload(sys)
sys.setdefaultencoding("utf-8")
from bs4 import BeautifulSoup
import requests
import pymysql

conn=pymysql.connect(host='localhost',user='sql_Weather',passwd='hello
sandy',db='sql_Weather',port=3306,charset='utf8')

```

```

cursor=conn.cursor()
sql="TRUNCATE TABLE Weather"
try:
    cursor.execute(sql)
    conn.commit()
except:
    print "Error: unable to delete"
def get_temperature(url):

    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132
Safari/537.36'} # 设置头文件信息
    response = requests.get(url, headers=headers).content # 提交
requests.get 请求
    soup = BeautifulSoup(response, "html.parser") # 用
Beautifulsoup 进行解析

    conmid = soup.find('div', class_='conMidtab')
    conmid2 = conmid.findAll('div', class_='conMidtab2')

    for info in conmid2:
        tr_list = info.find_all('tr')[2:] # 使用切片取到第三个 tr
        标签
        for index, tr in enumerate(tr_list): # enumerate 可以返回元
        素的位置及内容
            td_list = tr.find_all('td')
            if index == 0:
                province_name = td_list[0].text.replace('\n', '') #
                取每个标签的 text 信息, 并使用 replace() 函数将换行符删除
                city_name = td_list[1].text.replace('\n', '')
                weather = td_list[5].text.replace('\n', '')
                wind = td_list[6].text.replace('\n', '')
                max = td_list[4].text.replace('\n', '')
                min = td_list[7].text.replace('\n', '')
                print(province_name)
            else:
                city_name = td_list[0].text.replace('\n', '')
                weather = td_list[4].text.replace('\n', '')
                wind = td_list[5].text.replace('\n', '')
                max = td_list[3].text.replace('\n', '')
                min = td_list[6].text.replace('\n', '')

    #print(city_name, weather, wind, max, min)

```

```

        cursor.execute('insert into
Weather(city,weather,wind,max,min)
values(%s,%s,%s,%s,%s)', (city_name,weather,wind,max,min))
    if __name__=='__main__':
        urls = ['http://www.weather.com.cn/textFC/hb.shtml',
                'http://www.weather.com.cn/textFC/db.shtml',
                'http://www.weather.com.cn/textFC/hd.shtml',
                'http://www.weather.com.cn/textFC/hz.shtml',
                'http://www.weather.com.cn/textFC/hn.shtml',
                'http://www.weather.com.cn/textFC/xb.shtml',
                'http://www.weather.com.cn/textFC/xn.shtml']
        for url in urls:
            get_temperature(url)
    conn.commit()

```

四、用宝塔的 shell 脚本调用我们写的 py 脚本

我们设定 2 个小时爬一次（其实我感觉他更新并没有这么快）

脚本内容是我 py 文件的路径

