python爬虫开发记录（爬疫情数据）

本文的目的是来记录我们小组三人的开发记录，我们三人所编写的程序是爬取相关疫情 的数据，生成相应的表格形式的数据，最后进行可视化的操作。生成的疫情数据是可利用资源，供我们其他相关研究使用。以下将要开始讨论我们在开发过程中学习的记录，以及遇到的难点的记录。

开发过程中的学习的记录：

1. 网页构成

   了解了网页的一般构成：网页一般由三部分组成，分别是 **HTML**（超文本标记语言）、**CSS**（层叠样式表）和 **JavaScript**（简称"JS"动态脚本语言）。

2. 获取网页

   获取网页http信息并进行编解码操作：认识了第一个爬虫库：urllib，以及urlopen(),Request()方法的使用。例如：

```
def get_content_from_url(self, url):
    """
    根据URL获取响应内容
    :param url: 请求的URL
    :return: URL的响应的内容字符串
    """
    response = requests.get(url)
    return response.content.decode()
```

3. 用户代理池

   即访问网页的主机型号类型等，利用用户代理池是一种反爬技术。例如：

```
ua_list=[
'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Maxthon 2.0',
'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11
(KHTML, like Gecko) Chrome/17.0.963.56 Safari/535.11',
'User-Agent:Opera/9.80 (Windows NT 6.1; U; en) Presto/2.8.131
Version/11.11',
'Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1',
'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)',
'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50
(KHTML, like Gecko) Version/5.1 Safari/534.50',
'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0',
' Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1',
'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1',
' Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1)
Gecko/20100101 Firefox/4.0.1']
```

4. 正则表达式

在使用 Python 编写爬虫的过程中，re模块通常做为一种解析方法来使用。通过审查网页元素来获取网页的大体结构，然后使用解析模块来提取你想要的网页信息，最终实现数据的抓取。例如：

```
 json_str = re.findall(r'\[.+\]', text)[0]
```

5. 绘制生成表格：

```
xls = xlwt.Workbook()
sht = xls.add_sheet('CountryTypeService')
sht.write(0, 0, "国家")
sht.write(0, 1, "现存病例")
sht.write(0, 2, "累积病例")
sht.write(0, 3, "治愈数量")
sht.write(0, 4, "死亡数量")
l = 1
for country in countrys:
    sht.write(l, 0, country['provinceName'])
    sht.write(l, 1, country['currentConfirmedCount'])
    sht.write(l, 2, country['confirmedCount'])
    sht.write(l, 3, country['curedCount'])
    sht.write(l, 4, country['deadCount'])
    l += 1
xls.save("data/各国疫情数据.xls")
```

6. 可视化操作：

具体可视化的几个文件如下：

- dataVisualization/China_Covid_Map.ipynb
- dataVisualization/World_Covid_Map.ipynb
- dataVisualization/Top10_Covid_Province.ipynb
- dataVisualization/Top15_Covid_Countries.ipynb

本次操作主要是将疫情数据数据中的国家名字转换成我们可视化地图上的国家名字，并根据疫情数据来生成相应的国家颜色。

开发过程中的难点记录

1. 下载问题：

在编写python爬虫程序的时候，通常需要使用一些第三方库文件，有些库文件和python环境不匹配导致运行不成功。

2. 环境问题：

在编写python程序的时候通常需要调试好python，不然导致程序运行总出错或者干脆不运行。

3. 代码问题：

   basemap的参数数字不符合要，通过在网上寻找解决方案，找到数字进行替换成功解决问题。

4. 其他问题：

   爬取的网页源代码需要转换成Python文件，且需要多次确认数据类型。

5. 过程问题：

   可视化的建立部分