**计算机科学与工程学院实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验课程名称** | | **Python编程与数据分析** | | | **实验总成绩** |  |
| **专业** | **物联网工程** | | **班级** |  | **指导教师签字** |  |
| **学号** |  | | **姓名** | **陆晓晓** | **实验报告批改时间** |  |
| **实验报告分项成绩**   |  |  |  | | --- | --- | --- | | **序号** | **实验项目** | **成绩** | | **1** | **Project1** |  | | **2** |  |  | | **3** |  |  | | **4** |  |  | | **5** |  |  | | | | | | | |
| **实验课程总结**  从以下方面总结：1.实验体现知识应用和初步研究能力；2. 反映基本观察、发现问题和分析问题能力；3. 实验项目内容或者实验课程是否存在问题及下一年度改进意见）4.其他方面  这是第一次python实验，和李老师上课讲的内容有很大的不同，首先其实用到了许多上课没有学到的知识，需要自己去学习来实现。在这过程中，我认识到python与我之前学过的其他语言的异同。  Python 标准库和第三库众多，功能强大，这一点是c、c++比不了的，代码的封装程度很高，并且相关文档丰富，基本对应的库看一下文档就能会用了。在c、c++中100行的代码在python甚至只需要几行就能实现。  在写project1的时候我认识到自己对于python语言的练习还是不足的，之后会利用课后时间多加练习，争取把ppt上的例子都过一遍，这样也是为期末考试做准备。学好python也能为之后的科研、其他科目的作业、比赛打下很好的基础。 | | | | | | |

目录

[Project1 3](#_Toc12641)

[1. 实验目的 3](#_Toc8999)

[2. 实验需求分析 3](#_Toc4132)

[3. 实验内容 3](#_Toc10928)

[3.1实验运行环境 3](#_Toc32055)

[3.2程序设计过程思路、遇到问题 4](#_Toc19804)

[3.2.1数据预处理 4](#_Toc3676)

[3.2.2文件读写 4](#_Toc4471)

[3.2.3进度条实现 5](#_Toc23742)

[3.2.4获取首字母，如北京->BJ 7](#_Toc18510)

[3.2.5模糊查询 8](#_Toc31712)

[3.2.6写日志 8](#_Toc8983)

[3.2.7参数解析 10](#_Toc14751)

[4. 运行结果展示 11](#_Toc7087)

[5. 源代码 12](#_Toc26100)

[6. 心得体会 14](#_Toc7379)

# Project1

## 实验目的

加深对课上内容的理解，计划做一个关于二手房数据分析项目，共分4个小Project，每2周做一个。

项目需求为基本需求，没有特别的限定死，自由发挥的空间较大，使用的方法和库不设限。

## 实验需求分析

1）做一个控制台程序，输入省会城市名称，获取对应汉语拼音首字母，例：北京->BJ；

2）支持输入的模糊匹配（搜索“京”可以提示“北京”或者把“南京”和“北京”都匹配出来），带参数解析（参数为要加载城市文件的路径、要输出的日志路径）；

3）写日志（将日志写到log文件里，记录内容自定）；

4）任意方法做一个加载城市文件的进度条（城市文件自己编辑）

## 实验内容

### 3.1实验运行环境

引入库文件部分如下：

需要安装对应的第三方库：

1. **import** time #时间计算
2. **from** rich.progress **import** track #rich库实现进度条
3. **import** pinyin #另外使用pip安装pinyin包
4. **import** difflib #模糊搜索库
5. **import** logging #写日志
6. **import** numpy as np #构造一个小数的range
7. **import** sys #参数文件路径解析

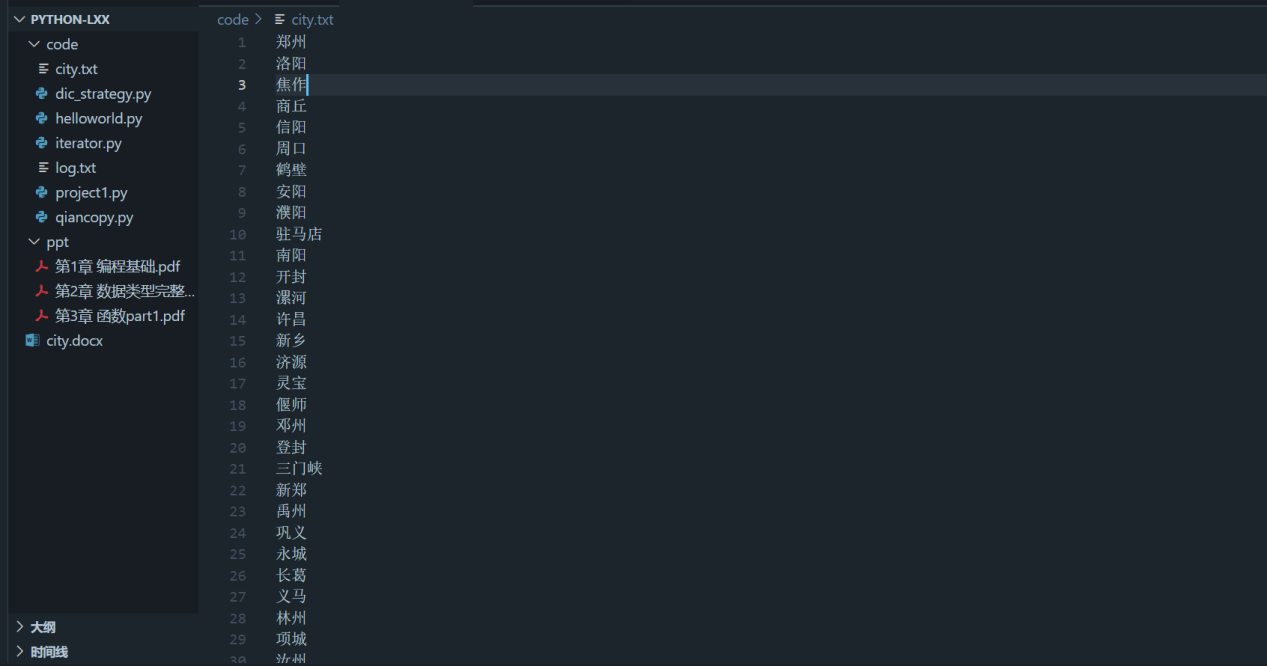
### 3.2程序设计过程思路、遇到问题

### 3.2.1数据预处理

首先，我们需要有一个城市的txt文件，里面是一些城市的名字。这个链接就是我找到的中国城市名称的概览。

https://zhidao.baidu.com/question/365761129.html

之后进行数据的预处理，将不需要的（如省会名）删除，按照老师给的例子，输入的应该是北京而不是北京市，所以也把所有的市删除，并且将所有城市整理为一行一个城市形式。如下图所示：



所有数据617条。

### 3.2.2文件读写

有了数据之后，我们要把这个数据读进来，并且存入list中，格式应该是：

[‘北京’,’天津’]，而不是[[‘北京’],[‘天津’]]，即这个城市list中，每个元素是str而不是list。

这个问题可以通过更换读取文件的方法，来解决。

造成这个问题的原因是一开始我为了图方便使用了spilt()，它会返回list。

1. **for** line **in** f:
2. txt.append((line.split(' '))

修改为如下的方法，可以解决这个问题，并且能够在读文件时候产生一个进度条。

1. txt=f.read()
2. **for** i **in** track(np.arange(0,0.0016,0.0001), description="Downloading..."):
3. str\_all = txt.split('\n')
4. time.sleep(0.000000001)

### 3.2.3进度条实现

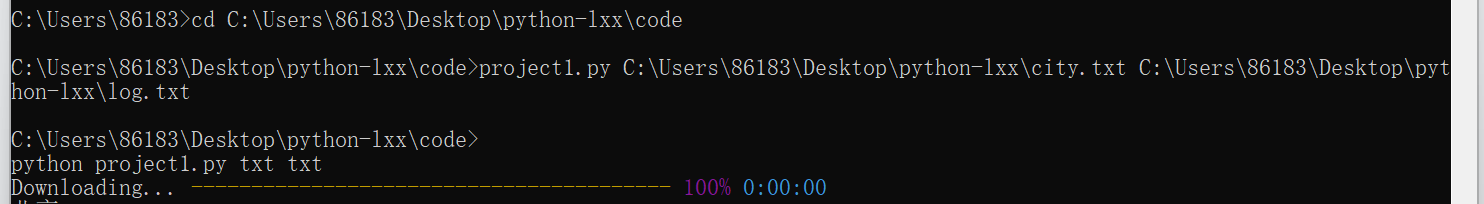
在这里实现的进度条，使用到了rich库，参考的教程如下：

https://geek-docs.com/python/python-3rd-packages/installing-and-using-rich-package-in-python.html

在Vscode中：效果：



在cmd中的效果：



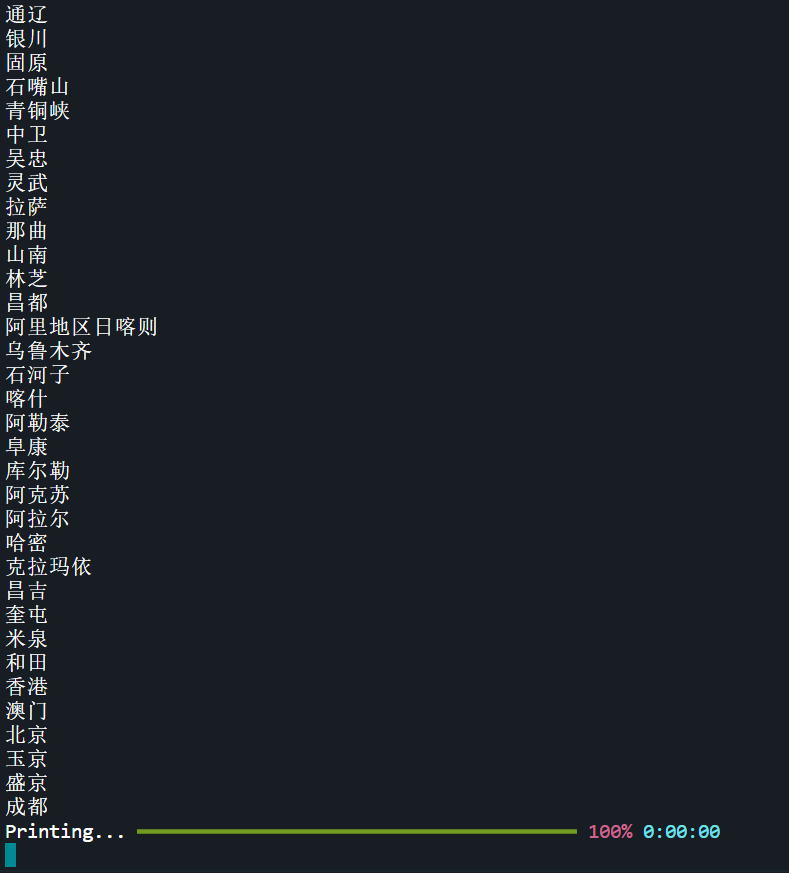
实现进度条部分的关键代码：

1. **for** i **in** track(np.arange(0,0.0016,0.0001), description="Downloading..."):
2. str\_all = txt.split('\n')
3. time.sleep(0.000000001)

主要依托track()函数实现的，这里np.arange()的含义可以类比range()产生一个序列来让for循环。

* 0是起始，
* 0.0016是终点，
* 0.0001是间隔，
* description是描述，就是这个进度条的名字。

另外，我还制作了一个打印所有城市的进度条：



作用是一边打印城市，一边有一个打印的进度条。

1. **for** i **in** track(range(len(str\_all)), description="Printing..."):
2. **print**(str\_all[i])
3. time.sleep(0.000000001)
4. #这是打印城市的进度条

用len获取list的长度，然后接着使用rich库的进度条。

### 3.2.4获取首字母，如北京->BJ

接下来是实现获取首字母的功能。

1. #这是使用pinyin库,获取输入文字的首字首字母拼音,输入文字的首字母拼音部分
3. **def** getStrAllAplha(str):
4. **return** pinyin.get\_initial(str, delimiter="").upper()
6. '''''
7. def getStrFirstAplha(str):
8. str=getStrAllAplha(str)
9. str=str[0:1]
10. return str.upper()
11. '''

14. str=input()
15. **print**(getStrAllAplha(str))
16. #print(getStrFirstAplha(str))

主要用到了pinyin库，

* delimiter 参数：可以设置两个拼音之间的分隔符。这里设置的是空，就是两个拼音之间没有分隔符。
* get\_initial()：可以帮助我们获取到拼音的首字母。
* upper() 方法将字符串中的小写字母转为大写字母。

返回字符串的第一个字的首字母也很简单，做一个切片就可以了。



### 3.2.5模糊查询

主要实现模糊查询的部分：

由于find函数的底层实现实际上是遍历list，加上数据量其实也不小，所以我使用了difflib库的get\_close\_matches()函数。

1. #这是模糊查询 find函数速度较慢
2. str2=input()
3. res = difflib.get\_close\_matches(str2, str\_all, 5, cutoff=0.6)
4. **print**(res)

该函数会返回最佳 “good enough” 匹配列表。

* word 是需要紧密匹配的序列；
* possibilities 是要匹配 word 的序列列表；
* 可选参数 n是要返回的最大匹配数；我这里使用的是5；
* 可选参数 cutoff (默认 0.6 )是 [0, 1] 范围内的浮点数。得分至少与word 相似的可能性将被忽略。我这里使用的是0.6就可以满足我们的需求了；

可能性中最好的(不超过n )匹配在列表中返回，按相似度得分排序，最相似的在前。

### 3.2.6写日志

写日志我使用的是：logging模块，相比print而言，logging可以通过设置不同的日志等级，在release版本中只输出重要信息，而不必显示大量的调试信息；

print将所有信息都输出到标准输出中，严重影响开发者从标准输出中查看其它数据；logging则可以由开发者决定将信息输出到什么地方，以及怎么输出；

日志代码部分：

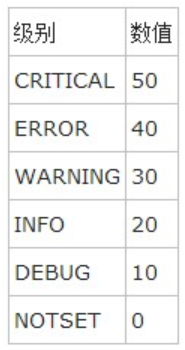
1. **def** writeln(path):
2. #写日志文件部分
3. logger = logging.getLogger(\_\_name\_\_)
4. logger.setLevel(level = logging.INFO)
5. handler = logging.FileHandler("log.txt")
6. handler.setLevel(logging.INFO)
7. formatter = logging.Formatter('%(asctime)s - %(project1)s - %(levelname)s - %(message)s')
8. handler.setFormatter(formatter)
9. logger.addHandler(handler)
11. logger.info("Start print log")
12. logger.debug("Do something")
13. logger.info("project1")
14. logger.error("ERROR message!")
15. logger.warning("abc")
16. logger.info("Finish your work!Good job!")

Formatters定义了最终log信息的内容格式，可以直接实例化Foamatter类。信息格式字符串用%(<dictionary key>)s风格的字符串做替换。

如何将日志输出到文件：

设置logging，创建一个FileHandler，并对输出消息的格式进行设置（我设置的是info，那么debug信息就不会输出），将其添加到logger，然后将日志写入到指定的文件中。

设置等级：setLevel(lvl)定义处理log的最低等级，内建的级别为：DEBUG，INFO，WARNING，ERROR，CRITICAL；下图是级别对应数值



这里有参考到博客：

https://www.cnblogs.com/qianyuliang/p/7234217.html#:~:text=logging%E6%A8%A1,%EF%BC%8C%E5%85%B7%E5%A4%87%E5%A6%82%E4%B8%8B%E4%BC%98%E7%82%B9%EF%BC%9A

设置好log输出的文件和信息之后，就基本完成了此次的功能。

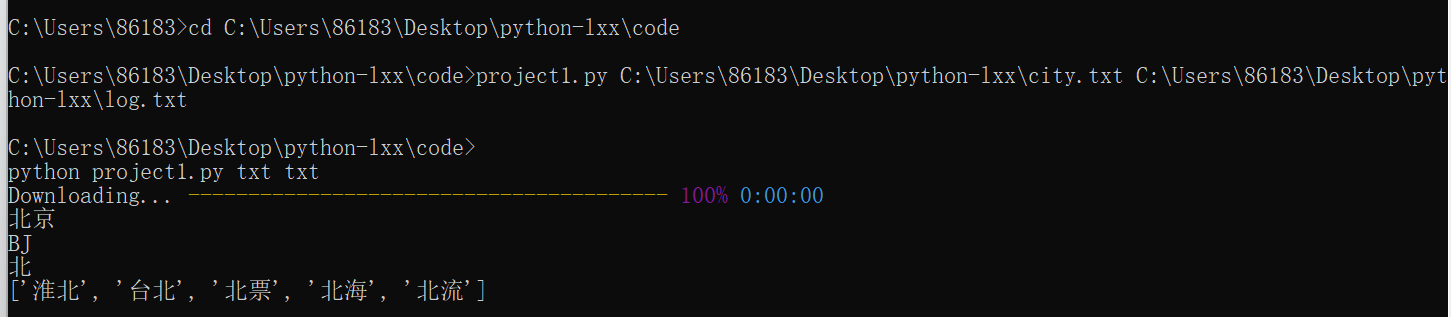
### 3.2.7参数解析

做作业到这里，其实我有些疑惑，之前从未接触过python，到底，什么是参数解析呢？通过上网看一些文档我大概理解了是在cmd中，可以使用python \*\*.py 参数1 参数2，这样运行代码。

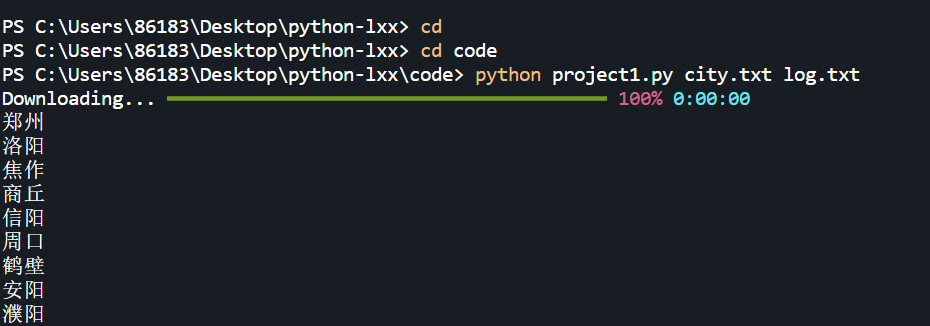
参数解析可以提高python脚本的复用性。这是跟我之前接触过c、c++的语言有很大不同的地方。

使用sys库，顺利完成了参数解析的设置。

这是在cmd中运行：



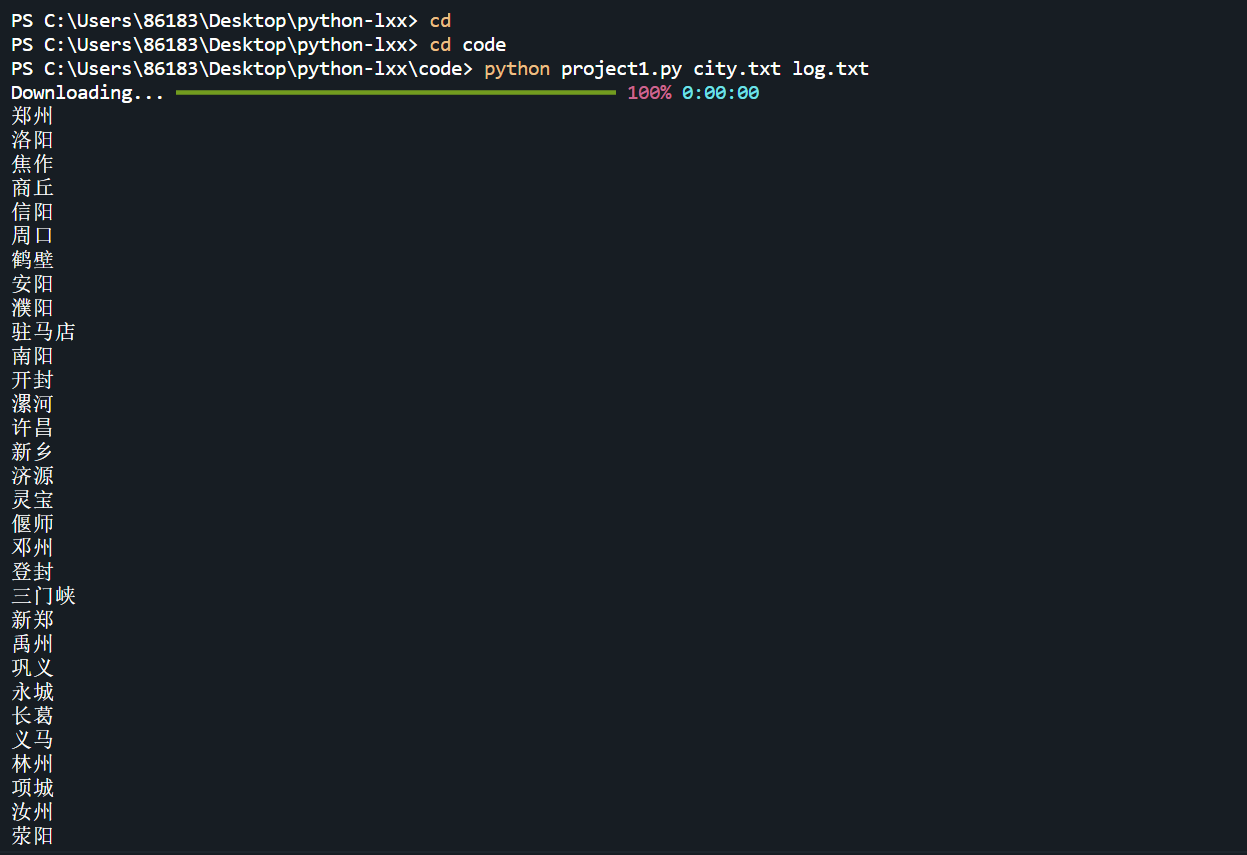
这是在vscode中运行：

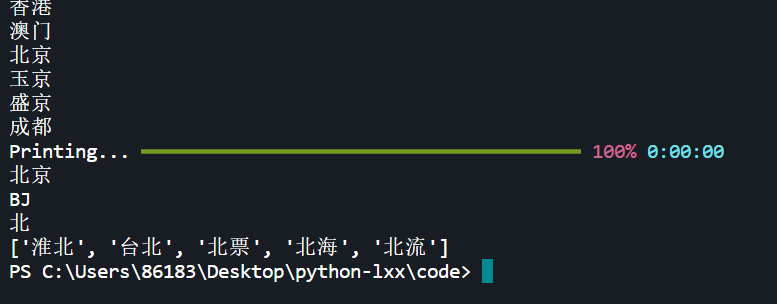


### 运行结果展示

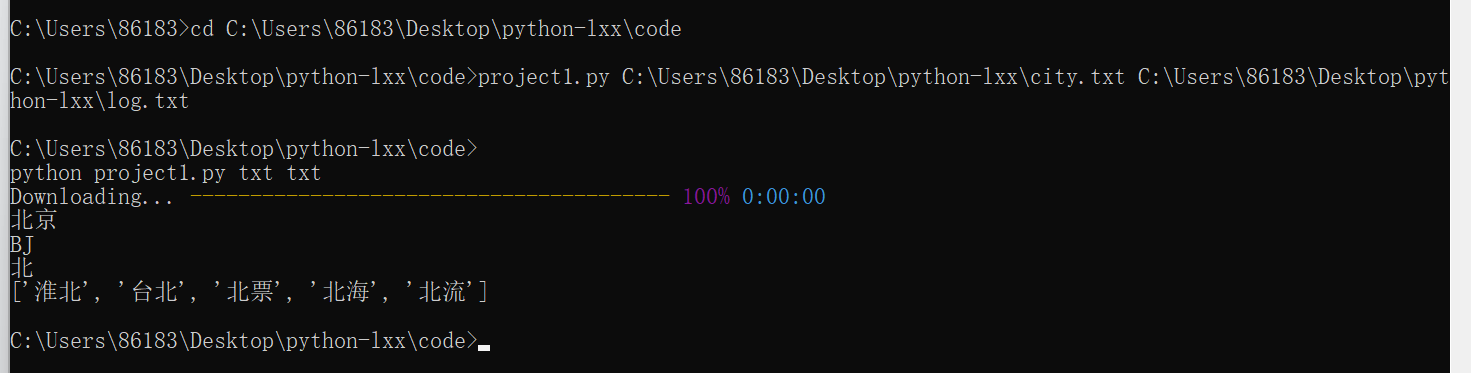
在vscode中，首先要进入code 的文件夹：

然后输入python project1.py city.txt log.txt回车：





在cmd 中，也要先使用cd进入文件夹才行。



### 源代码

1. **import** time #时间计算
2. **from** rich.progress **import** track #rich库实现进度条
3. **import** pinyin #另外使用pip安装pinyin包
4. **import** difflib #模糊搜索库
5. **import** logging #写日志
6. **import** numpy as np #构造一个小数的range
7. **import** sys #参数文件路径解析
9. #读文件部分
10. with open('city.txt','r',encoding='utf-8') as f:
11. '''''for line in f:
12. txt.append((line.split(' '))'''
14. #       time\_start = time.time() #计时0.0016s\
15. txt=f.read()
16. **for** i **in** track(np.arange(0,0.0016,0.0001), description="Downloading..."):
17. str\_all = txt.split('\n')
18. time.sleep(0.000000001)
19. #    time\_end = time.time()
20. #time\_c= time\_end - time\_start   #运行所花时间
21. #print('time cost', time\_c, 's')
22. #print(str\_all)
23. f.close()
25. '''''
26. for i in track(range(len(str\_all)), description="Printing..."):
27. print(str\_all[i])
28. time.sleep(0.000000001)
29. #这是打印城市的进度条
30. '''
32. #这是使用pinyin库,获取输入文字的首字首字母拼音,输入文字的首字母拼音部分
34. **def** getStrAllAplha(str):
35. **return** pinyin.get\_initial(str, delimiter="").upper()
37. '''''
38. def getStrFirstAplha(str):
39. str=getStrAllAplha(str)
40. str=str[0:1]
41. return str.upper()
42. '''

45. str=input()
46. **print**(getStrAllAplha(str))
47. #print(getStrFirstAplha(str))


51. #这是模糊查询 find函数速度较慢
52. str2=input()
53. res = difflib.get\_close\_matches(str2, str\_all, 5, cutoff=0.6)
54. **print**(res)

57. **def** writeln(path):
58. #写日志文件部分
59. logger = logging.getLogger(\_\_name\_\_)
60. logger.setLevel(level = logging.INFO)
61. handler = logging.FileHandler("log.txt")
62. handler.setLevel(logging.INFO)
63. formatter = logging.Formatter('%(asctime)s - %(project1)s - %(levelname)s - %(message)s')
64. handler.setFormatter(formatter)
65. logger.addHandler(handler)
67. logger.info("Start print log")
68. logger.debug("Do something")
69. logger.info("project1")
70. logger.error("ERROR message!")
71. logger.warning("abc")
72. logger.info("Finish your work!Good job!")

75. **if** \_\_name\_\_ == '\_\_main\_\_':
76. argv=sys.argv
77. city\_str=argv[1]
78. log\_str=argv[2]

### 心得体会

这是第一次python实验，和李老师上课讲的内容有很大的不同，首先其实用到了许多上课没有学到的知识，需要自己去学习来实现。在这过程中，我认识到python与我之前学过的其他语言的异同。

Python 标准库和第三库众多，功能强大，这一点是c、c++比不了的，代码的封装程度很高，并且相关文档丰富，基本对应的库看一下文档就能会用了。在c、c++中100行的代码在python甚至只需要几行就能实现。

在写project1的时候我认识到自己对于python语言的练习还是不足的，之后会利用课后时间多加练习，争取把ppt上的例子都过一遍，这样也是为期末考试做准备。学好python也能为之后的科研、其他科目的作业、比赛打下很好的基础。