**云南大学软件学院**

**实 验 报 告**

课程： Python语言设计实践 任课教师： 胡盛 实验指导教师： 胡盛

姓名： 孙启奥 学号： 20171120028 专业： 软件工程 日期： 成绩：

**实验五 组合数据类型综合应用 – 记忆字典**

1. **实验目的**

熟悉字典、列表、元组、列表、字符串、文件等组合数据类型，学习界面图像化设计和数据持久化操作，并依据具体需求进一步践行CDIO（Conceive：构思，Design：设计，Implement：实现和Operate：运作）的工程理念。

**二、****实验要求**

（1）必须使用Python3.X版本.

（2）使用Python自带的IDLE或PyCharm为开发工具.（自行安装）

（3）请自行完成实验内容，切勿抄袭.

（4）编码要规范, 并且核心代码有注释，说明其处理意图,

（5）若使用第三方库必须做详细介绍和说明，并提供下载地址.

（6）提交实验报告：

电子版：电子版（doc版本）实验报告，按时间要求上传到云南大学软件学院实验教学网http://seilab.ynu.edu.cn.

**三、实验内容**

A.（参考难点系数0.7） 完成一个基本的字典程序，具备如下功能：

1. 字典词库至少有10个不同的英语单词，每个单词有拼写、词性（n，v，adj，adv等）、中文释义和典型例句；
2. 添加 – 可向字典词库添加一个单词，单词拼写（必填）、词性（必填）、中文释义（必填）、典型例句（可选）；
3. 删除 – 可删除字典词库中的一个单词；
4. 更新 – 可以更新字典词库中一个单词的基本信息 ；
5. 查询 – 输入单词拼写，可以查询单词相关的基本信息；
6. 浏览 – 可以按照逆序或者升序浏览字典词库中的单词；
7. 保存 – 字典词库信息更改后能够保存，下次能够读取新的字典词库信息。

B.（参考难点系数0.8） 在A的基础上，通过图形化的操作界面完成上述操作，并添加一个可以自定义模板导入导出字典词库的功能。

C.（参考难点系数0.9） 在B的基础上，不局限上述功能完成一个你理解的智能记忆字典。

**四、设计、实现与分析（设计给出你的设计思想，实现要有截图和文字说明，分析要有实验结果分析和总结，集成他人的成果要注明并给出出处）**

**1.主要设计思想：**

在本次实验中，我主要设计了一个可以用于单词查询以及单词查询情况记录还有单词复习功能的桌面程序字典。应该属于C难度，当时没考虑可以使用web前端程序来进行设计使用，所以做出来可能看上去就不是很漂亮。但是基本功能都可以实现，基于Pymongo，Easygui，做的字典前后端，数据库选择了时下比较流行的非关系型数据库MongoDB，数据集来源是自己通过Requests加BeautifulSoup库进行爬取的，数量不是很多，一共十页，大概在200个单词左右，如果想要更多可以自己设置页数重新爬取。

**2.主要使用库介绍：**

**1.前后端库**

**（1）Pymongo:**Pymongo是Python中用来操作MongoDB的一个库。而MongoDB是一个基于分布式文件存储的数据库，旨在为WEB应用提供可扩展的高性能数据存储解决方案。其文件存储格式类似于JSON，叫BSON，不严谨、通俗地理解，就是Python中的字典键值对格式。所以要利用Python操作MongoDB，有以下两个步骤：1.在电脑上安装MongoDB；2.在Python上装入pymongo的库。

在网络上有很多关于Pymongo的介绍，所以在这里就不介绍具体的使用方法了

**官方文档网站：**

<https://pymongo.readthedocs.io/en/stable/>

**(2).EasyGui:** EasyGUI是使用Python进行非常简单，非常容易的GUI编程的模块。EasyGUI与其他GUI生成器的不同之处在于，EasyGUI不是事件驱动的。相反，所有GUI交互都通过简单的函数调用来调用。EasyGui提供了一个易于使用的界面，用于与用户进行简单的GUI交互。它不需要程序员了解有关tkinter，框架，小部件，回调或lambda的任何信息。EasyGUI在Python 2和3上运行，并且没有任何依赖关系。

**官方文档网站：**

<https://pythonhosted.org/easygui/index.html>

**2.爬虫库介绍：**

**（1）Requests:** 更好的方案是使用requests。它是一个Python第三方库，处理URL资源特别方便。引用 Requests 官网的说明：Requests is the only Non-GMO HTTP library for Python, safe for human consumption.Requests 官方的介绍语是多么霸气。之所以能够这么霸气，是因为 Requests 相比 urllib 在使用方面上会让开发者感到更加人性化、更加简洁、更加舒服。

**使用方法文档：这是廖雪峰老师的介绍：**

<https://www.liaoxuefeng.com/wiki/1016959663602400/1183249464292448>

**（2）Beautifulsoup：**

Beautiful Soup是一个可以从HTML或XML文件中提取数据的Python库，简单来说，它能将HTML的标签文件解析成树形结构，然后方便地获取到指定标签的对应属性。

通过Beautiful Soup库，我们可以将指定的class或id值作为参数，来直接获取到对应标签的相关数据，这样的处理方式简洁明了。

**官方文档：**

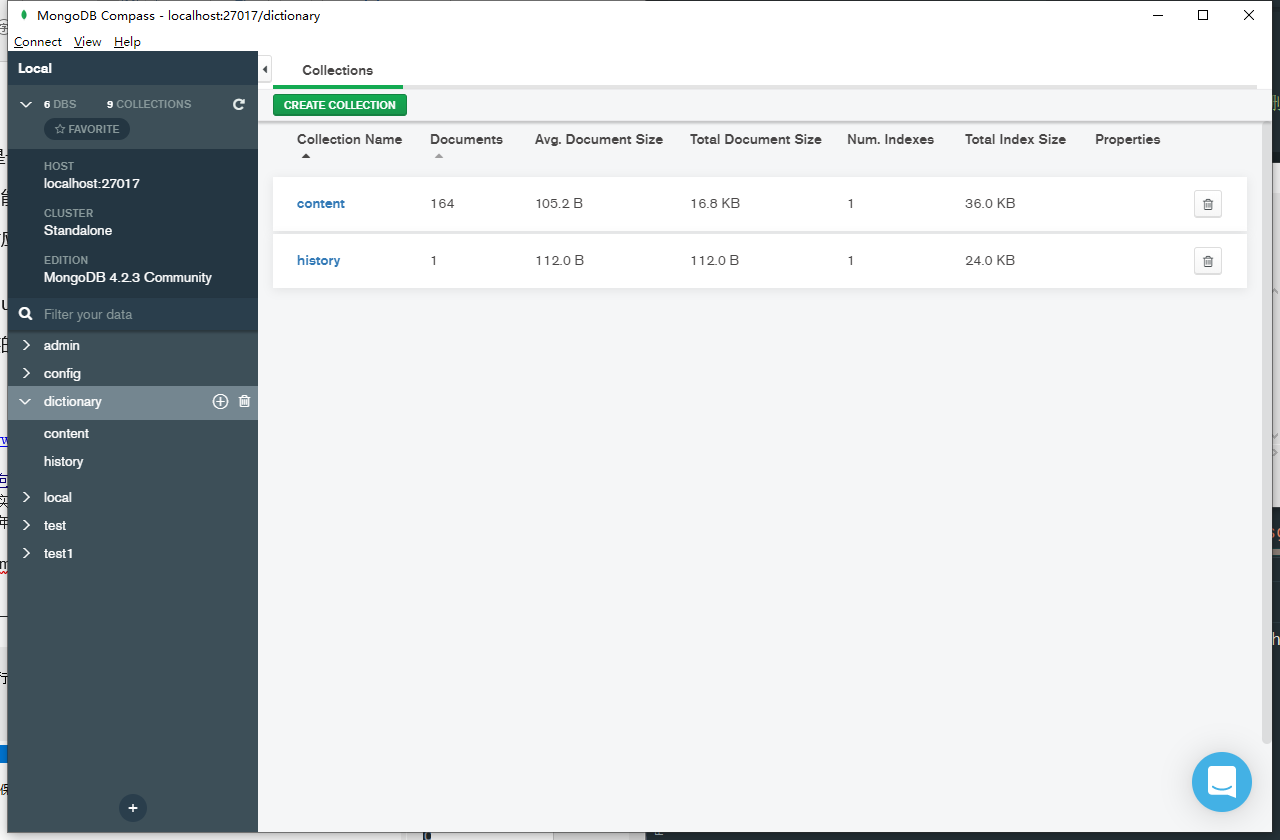
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/index.zh.html>

**3.MongoDB介绍：**

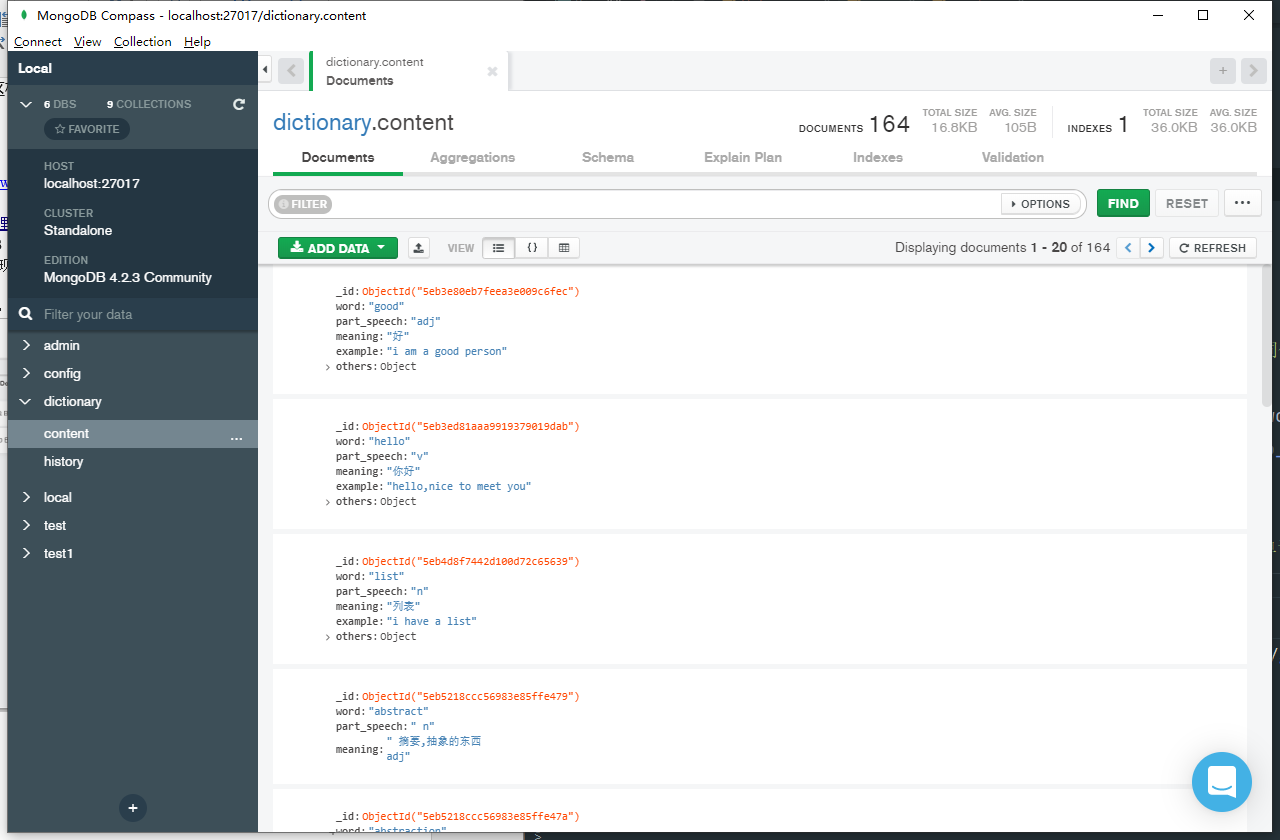
MongoDB是一种[面向文档](https://zh.wikipedia.org/wiki/%E9%9D%A2%E5%90%91%E6%96%87%E6%AA%94%E7%9A%84%E6%95%B8%E6%93%9A%E5%BA%AB" \o "面向文档的数据库)的[数据库管理系统](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)，用[C++](https://zh.wikipedia.org/wiki/C%2B%2B)等语言撰写而成，以解决应用程序开发社区中的大量现实问题。MongoDB由[MongoDB Inc.](https://zh.wikipedia.org/w/index.php?title=MongoDB_Inc.&action=edit&redlink=1" \o "MongoDB Inc.（页面不存在）)（当时是[10gen](https://zh.wikipedia.org/w/index.php?title=10gen&action=edit&redlink=1" \o "10gen（页面不存在）)团队）于2007年10月开发，2009年2月首度推出，现以[服务器端公共许可](https://zh.wikipedia.org/w/index.php?title=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF%E5%85%AC%E5%85%B1%E8%AE%B8%E5%8F%AF&action=edit&redlink=1" \o "服务器端公共许可（页面不存在）)（SSPL）分发。

**4.数据库设计**

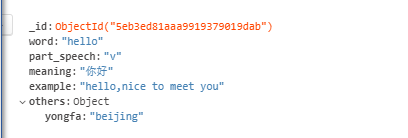
本次实验，我使用了mongoDB数据库，建立了两个Collection，如图所示：



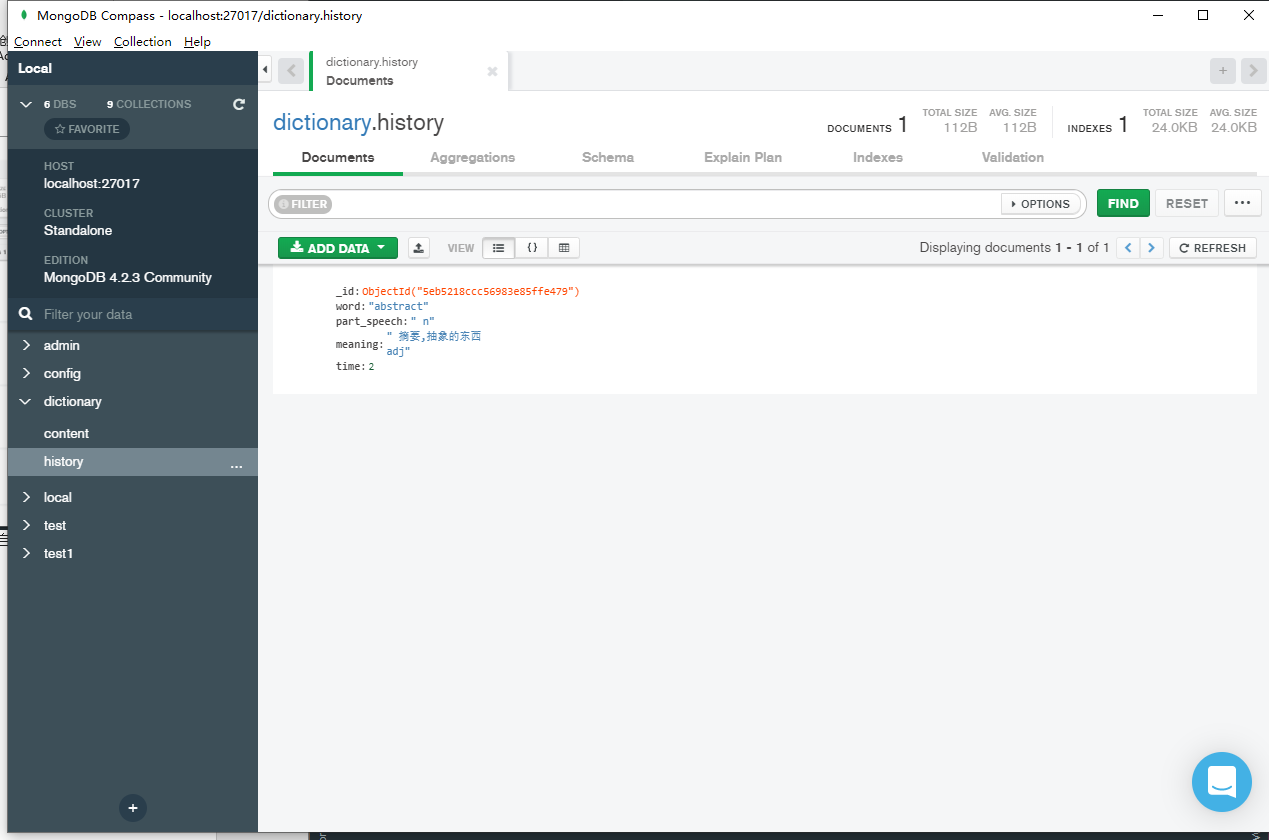
其中第一个collection是字典内容，格式及内容如图所示：



其中包括每一个单词的词义，词性，意思，例句以及其他自定义部分，自定义部分以字典呈现：

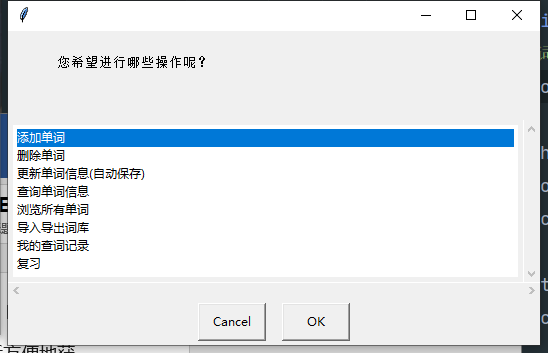


除此之外，历史查询部分我建立了一个新的collection，如图所示:



其中包括单词，词性，词义，以及查询次数，当你在对一个单词查询多次的时候，会更改查询次数的属性time.

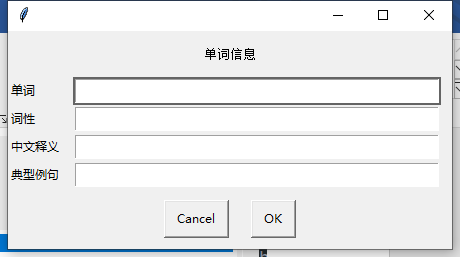
**5.实现效果**



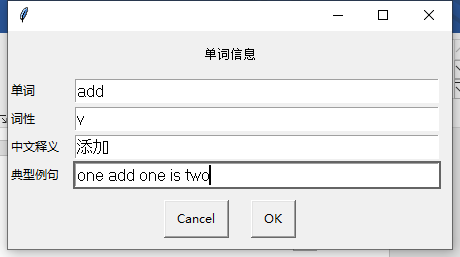
所有操作种类：

1. 添加单词：

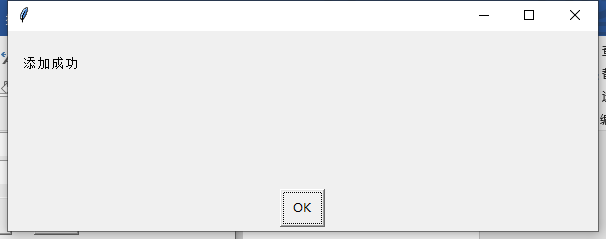
选择添加单词之后的界面：



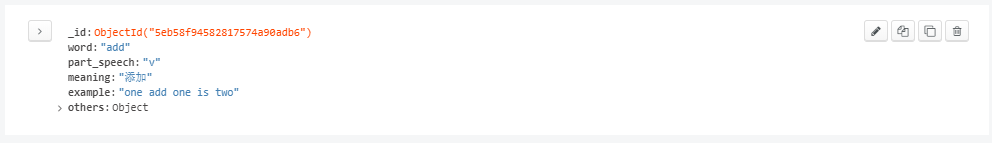
增加单词：



OK之后：

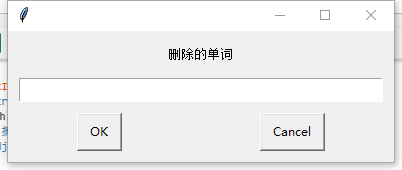


之后查看mongodb：

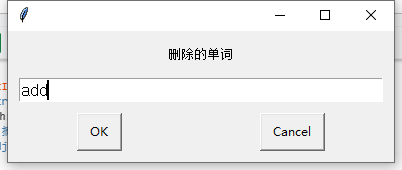


发现已经添加成功了。

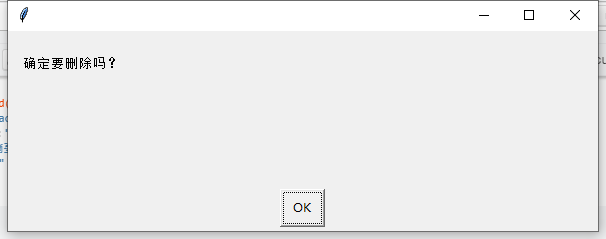
1. 删除单词：

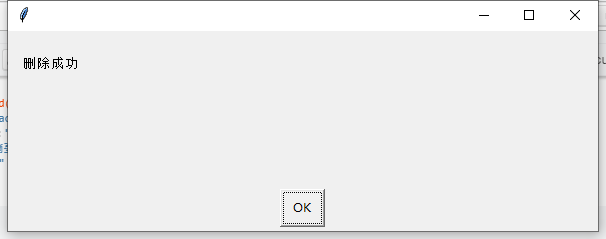


选择之后，可以输入需要删除的单词，以add举例，

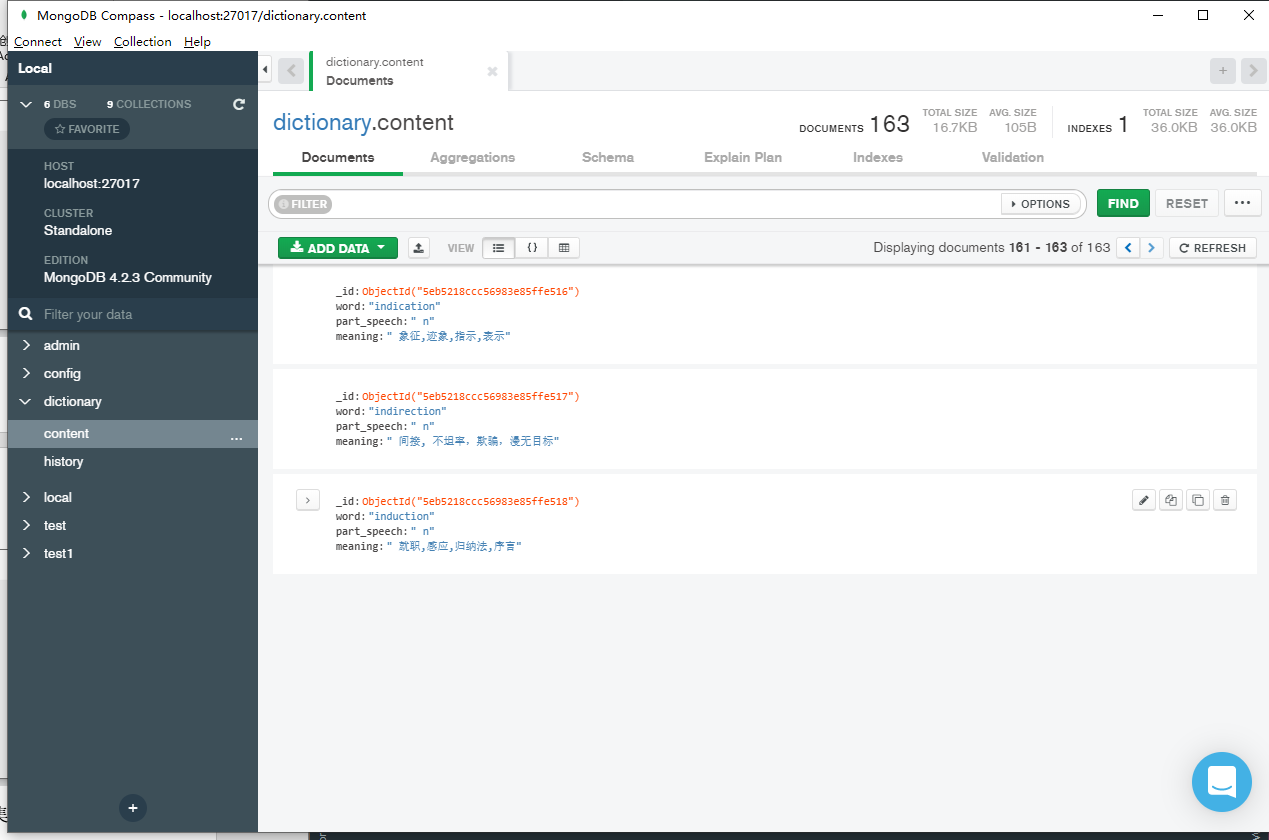


之后会要求确定：



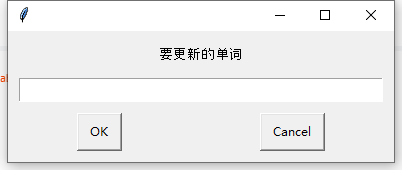


查看数据库：

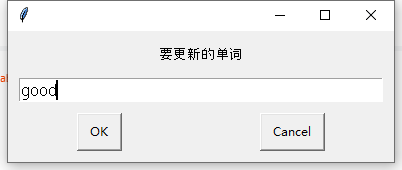


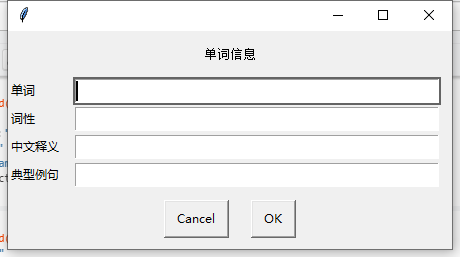
发现已经删除成功了。

1. 更新单词：

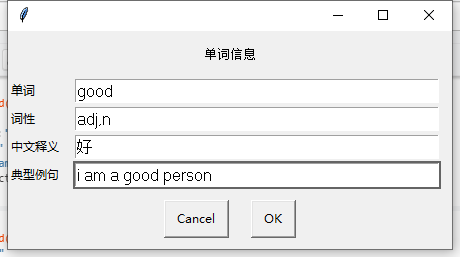


此处以good为例子：

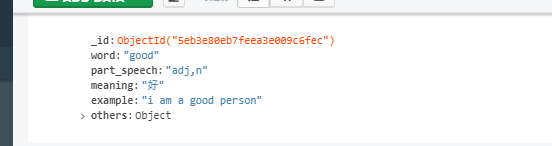




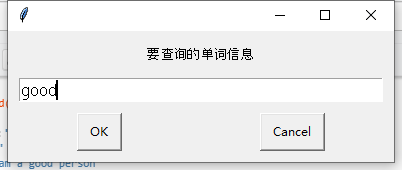
需要写出想更改的部分。



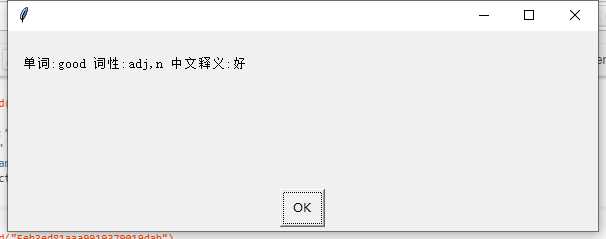
查看数据库，发现已经更新成功了：（将adj更改为adj,n）



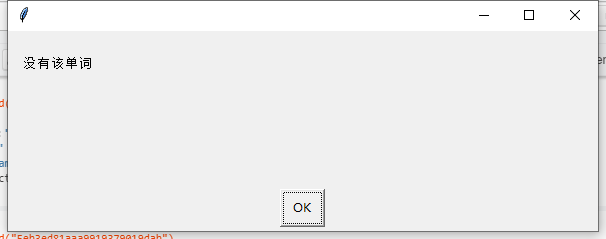
4.查询



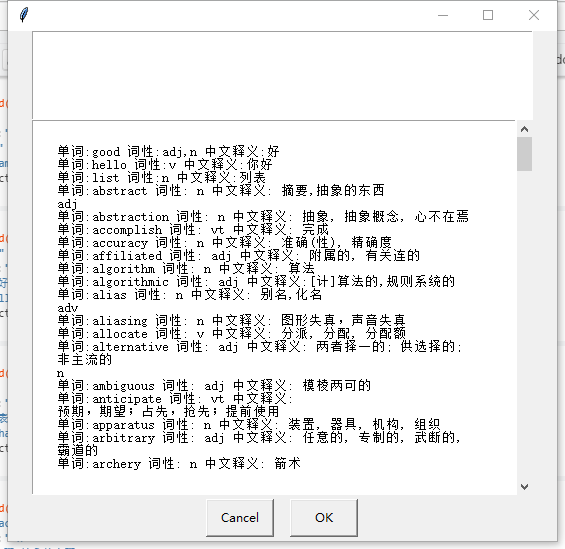
查询结果展示：



若没有查询到，则展示：

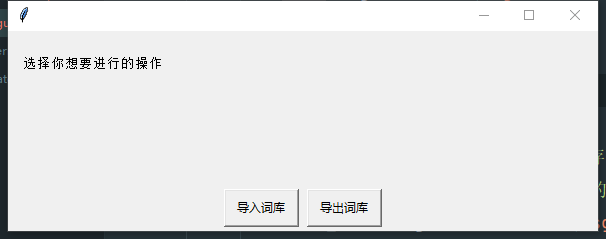


1. 之后展示全部单词：



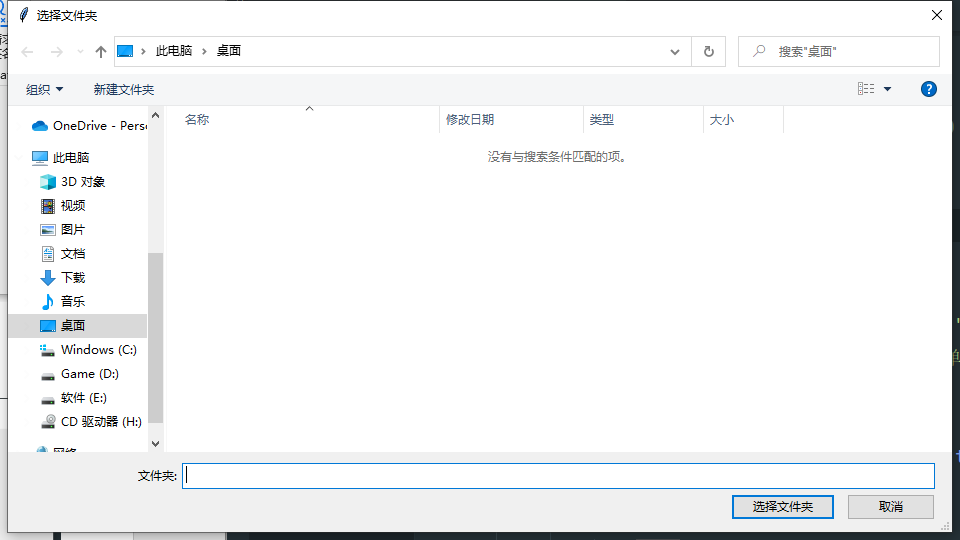
已经全部展示出来了。

1. 导出词库：

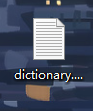


选择导出词库：

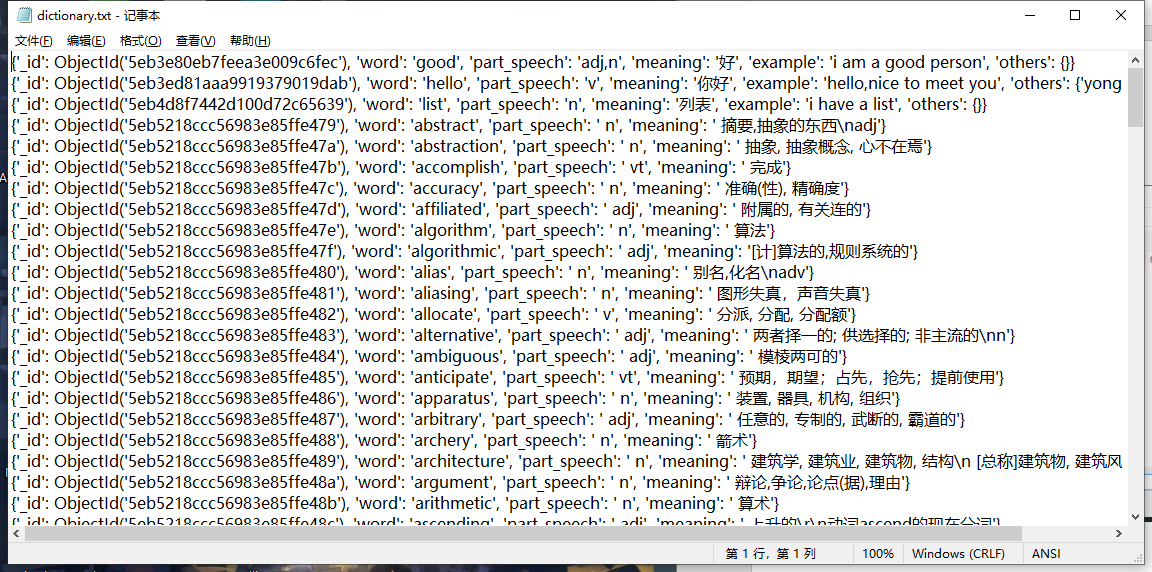
选择保存在桌面：



发现已经保存成功了：



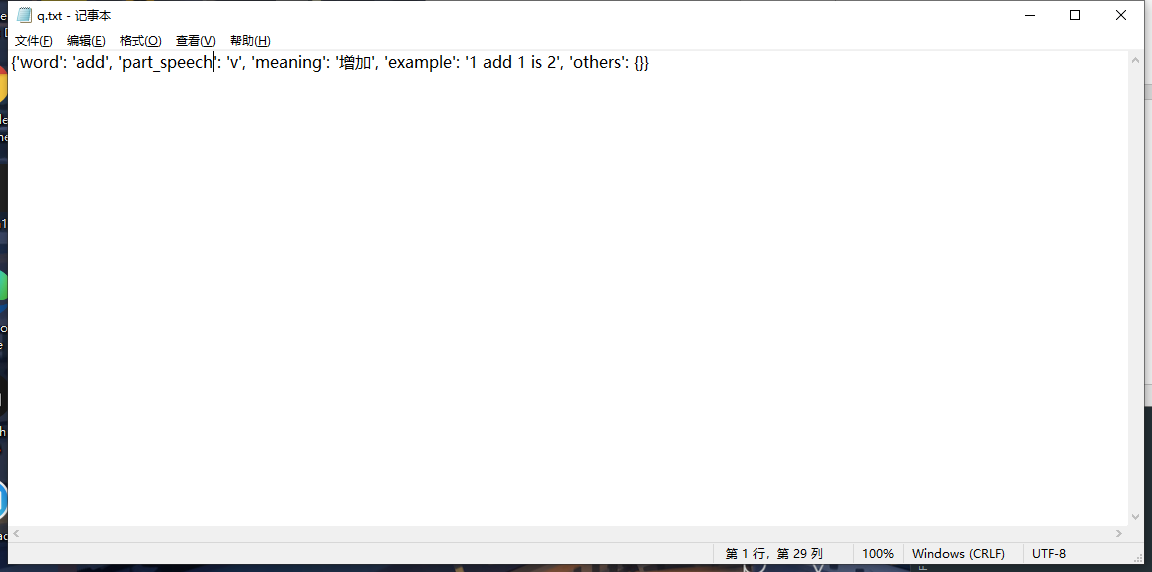
内容如下：



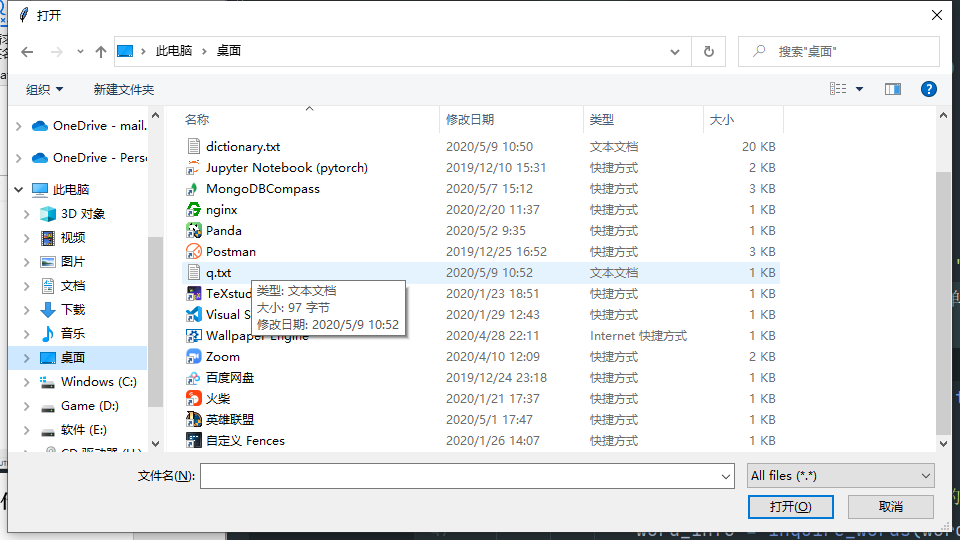
导出为json格式。

1. 导入词库：

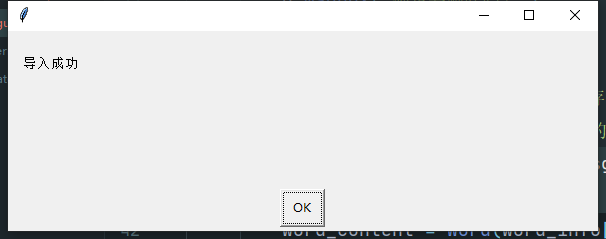
导入格式也设定的是dict格式：



选择需要导入的词库文件：



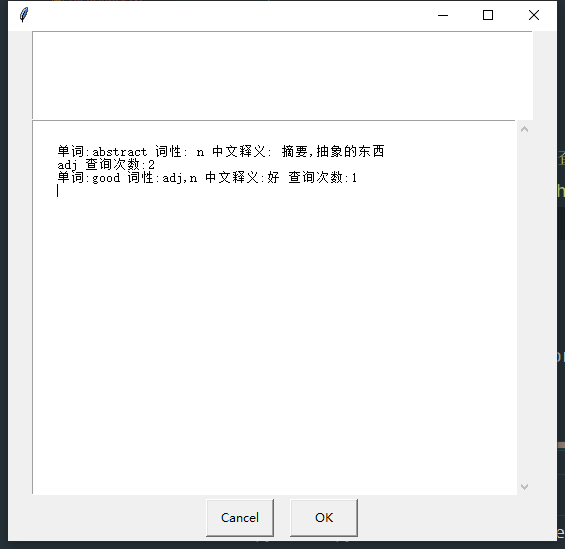
只要格式正确就会导入成功。



查看数据库，发现已经增加成功了。



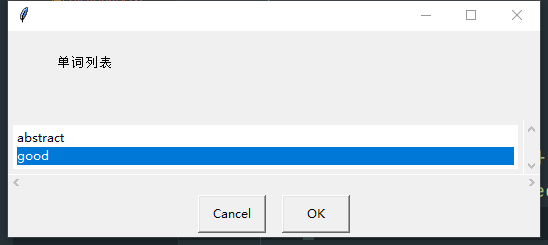
1. 可以查询我的查词记录：



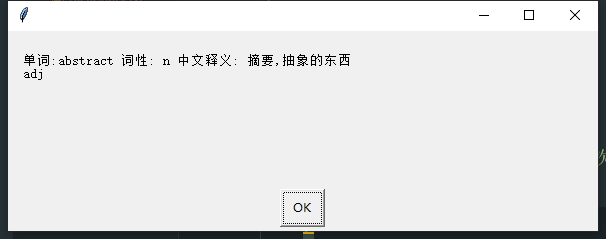
可以知道每个单词查询了多少次，还是比较方便的。

1. 复习功能，可以查到自己搜索过的所有单词，然后不显示词义，当你不知道词义的时候，可以双击显示单词信息，方便记忆。

单词列表如图所示：



双击abstract：



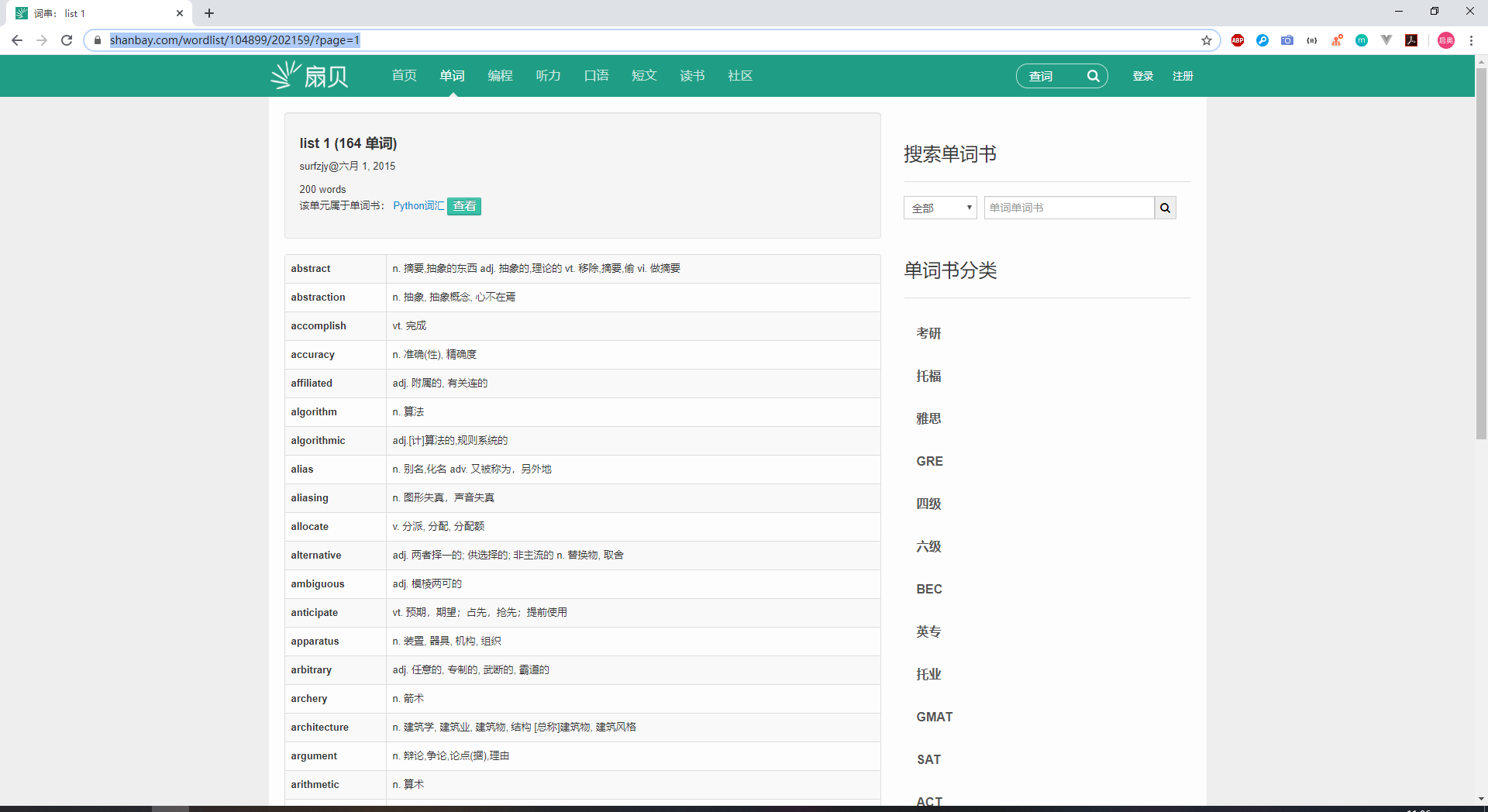
可以显示abstract的信息。

以上所有字典设计及演示就全部完成了。

之后简要介绍一下使用的爬虫。

本次所要爬取的页面是：

<https://www.shanbay.com/wordlist/104899/202159/?page=1>

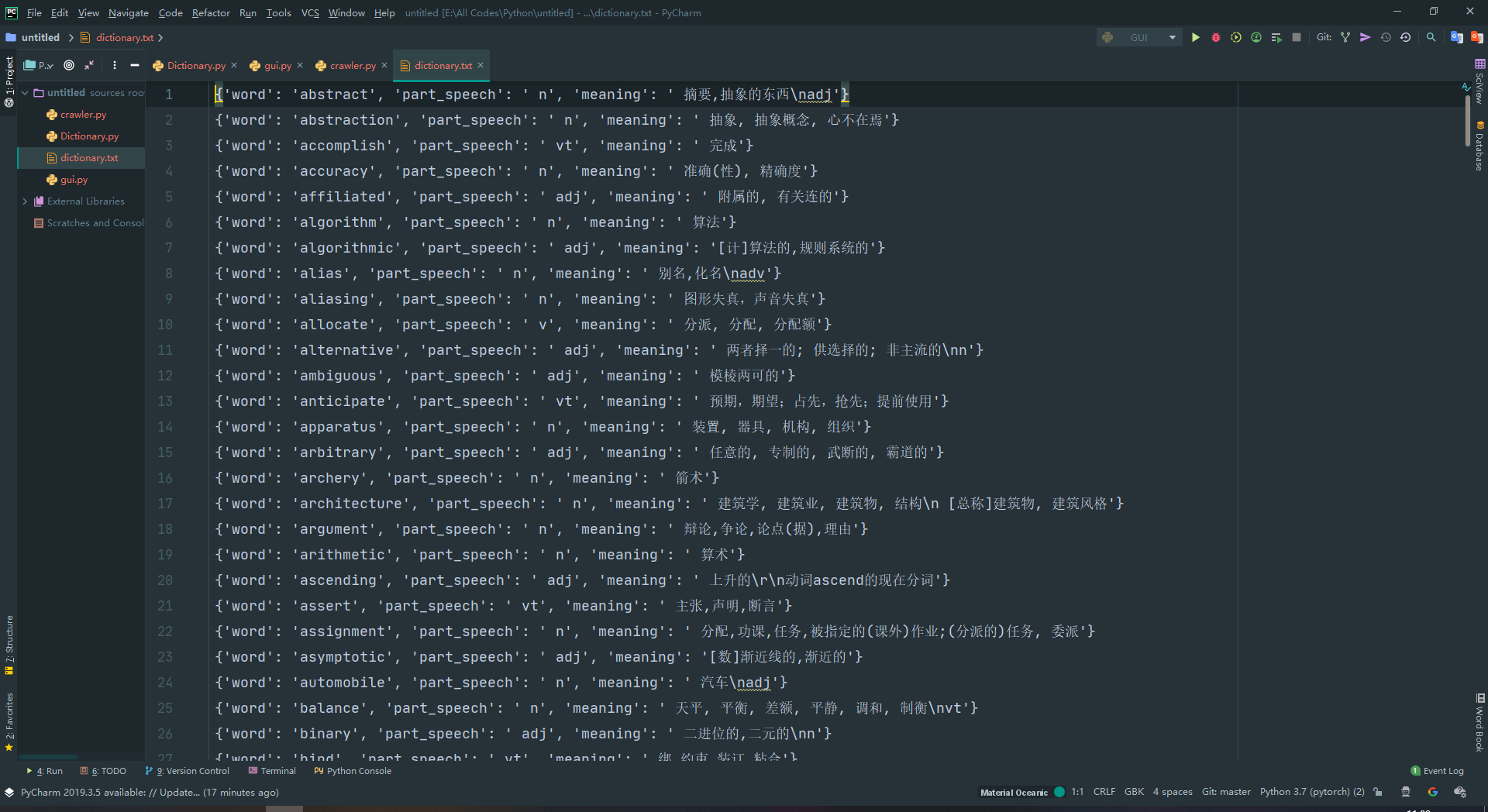


选取这个页面的主要原因是因为单词较全，且格式与需要的格式类似。

使用request进行访问并分析页面源代码，并使用bs4进行网页源代码分析。

爬取结果使用字典存储。

查询结果展示。

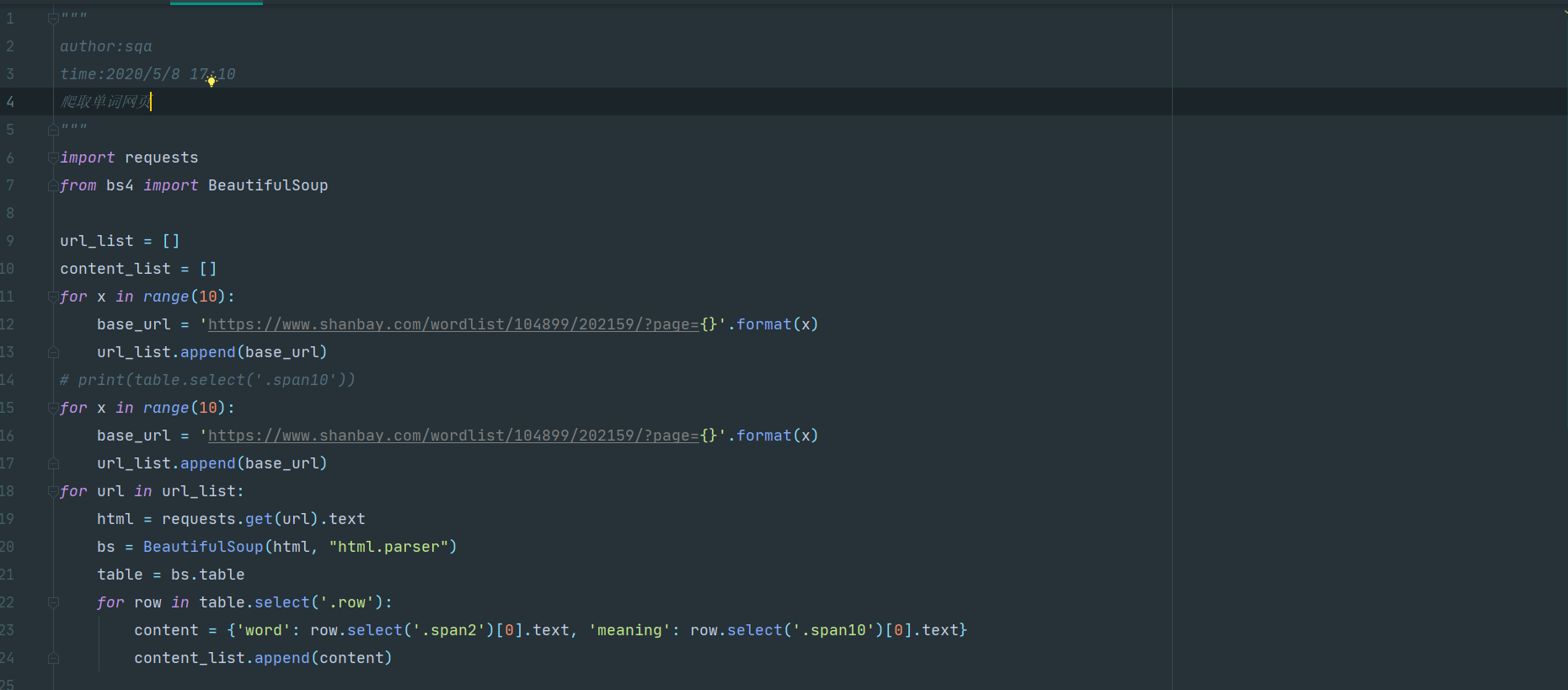


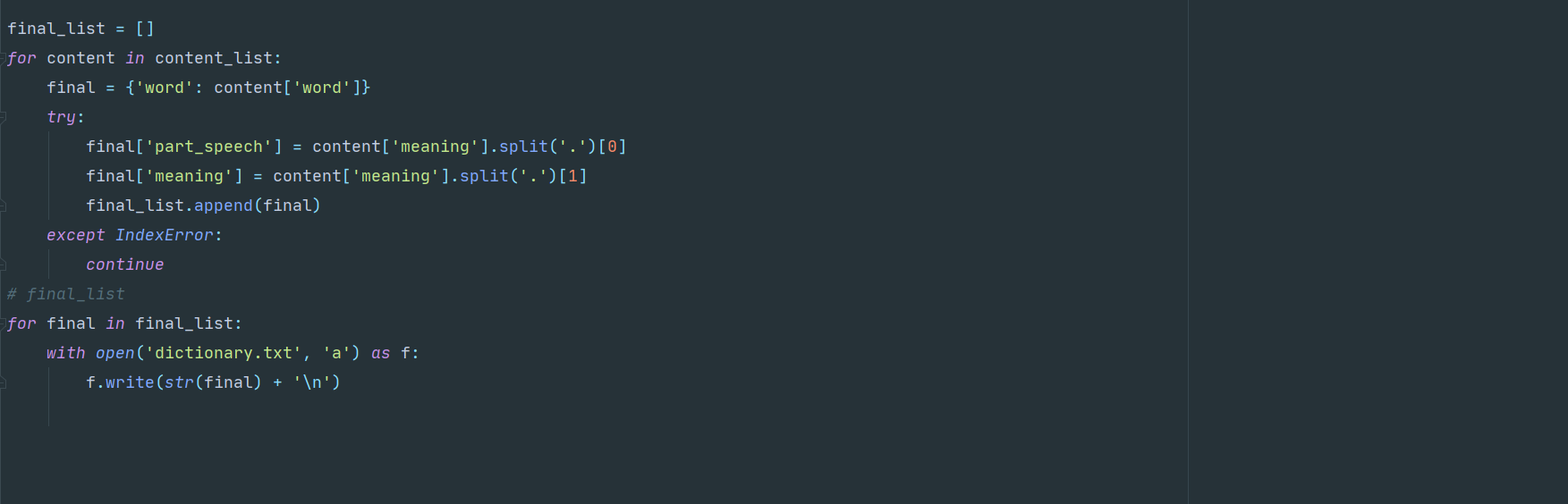
**五、附录（完整源码，注意编码风格，关键代码要注释，集成他人的代码要注明并给出出处）**

温馨提示：前后文字字体、大小要统一

代码：

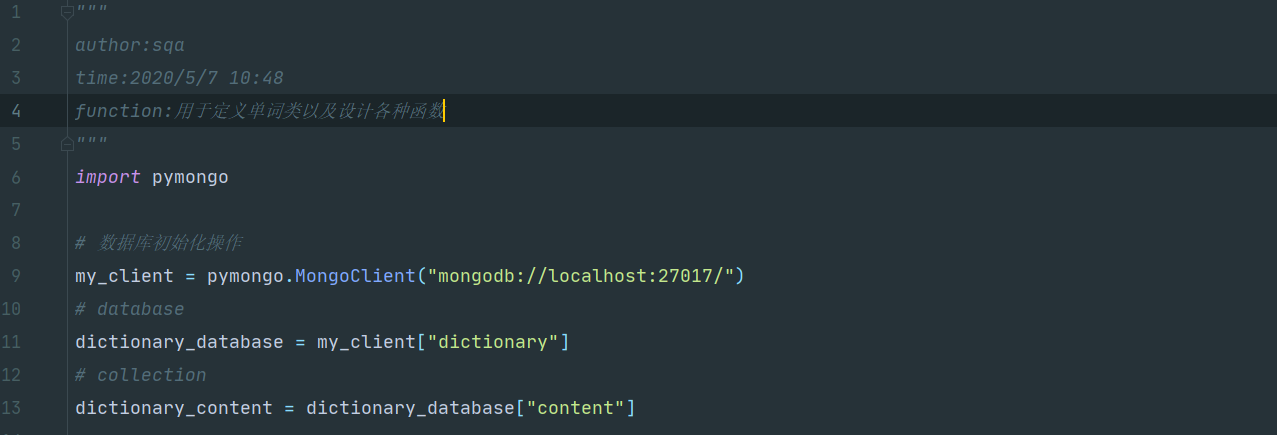
1. crawler.py(爬虫程序)





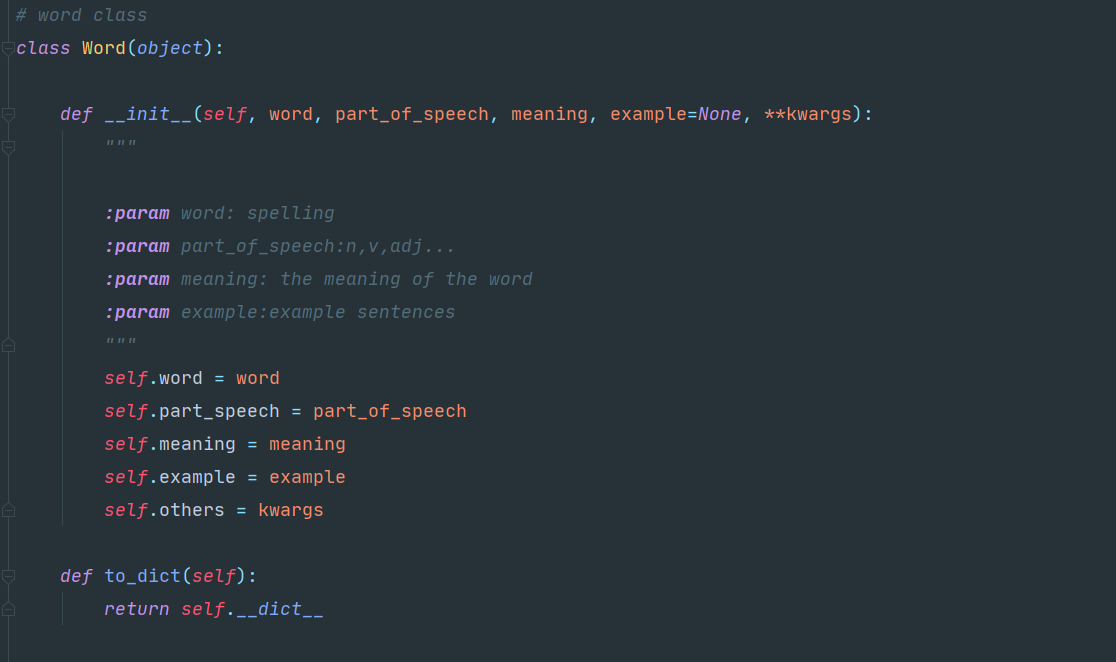
1. **Dictionary.py:**

**数据库初始化**



**单词类：**

**包含一个把该类的属性通过字典表示的方法**



**删除单词的方法:**



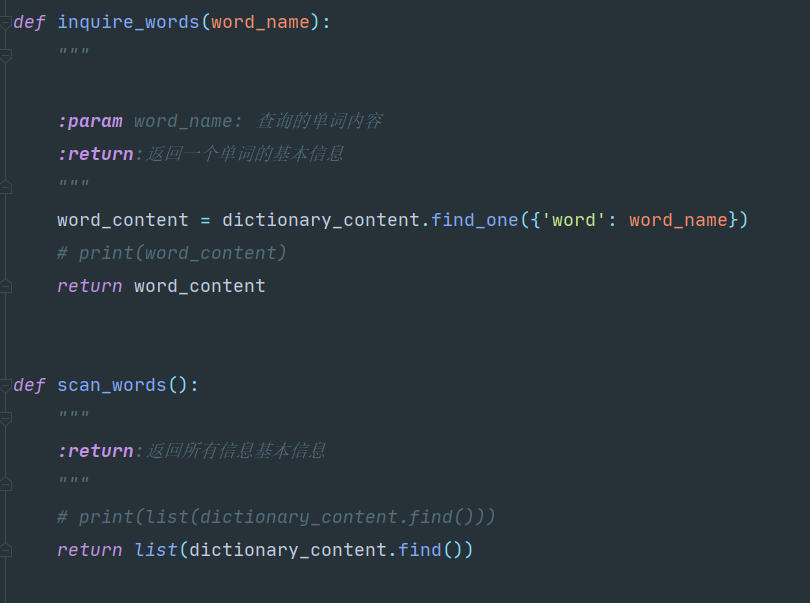
**添加单词的方法：**



**更新单词信息的方法：**

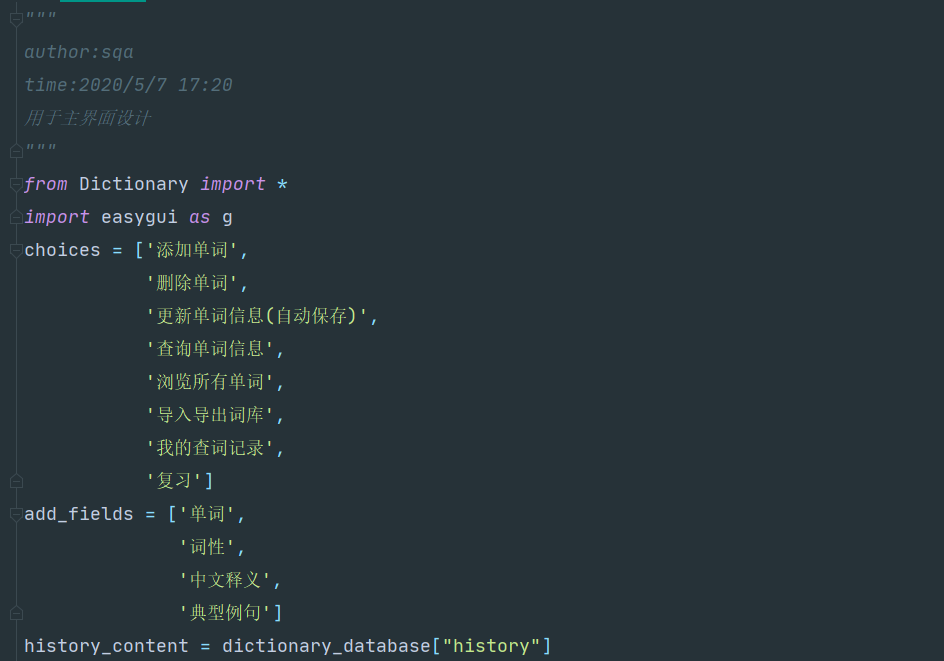


**查询单词和查询所有单词的方法：**

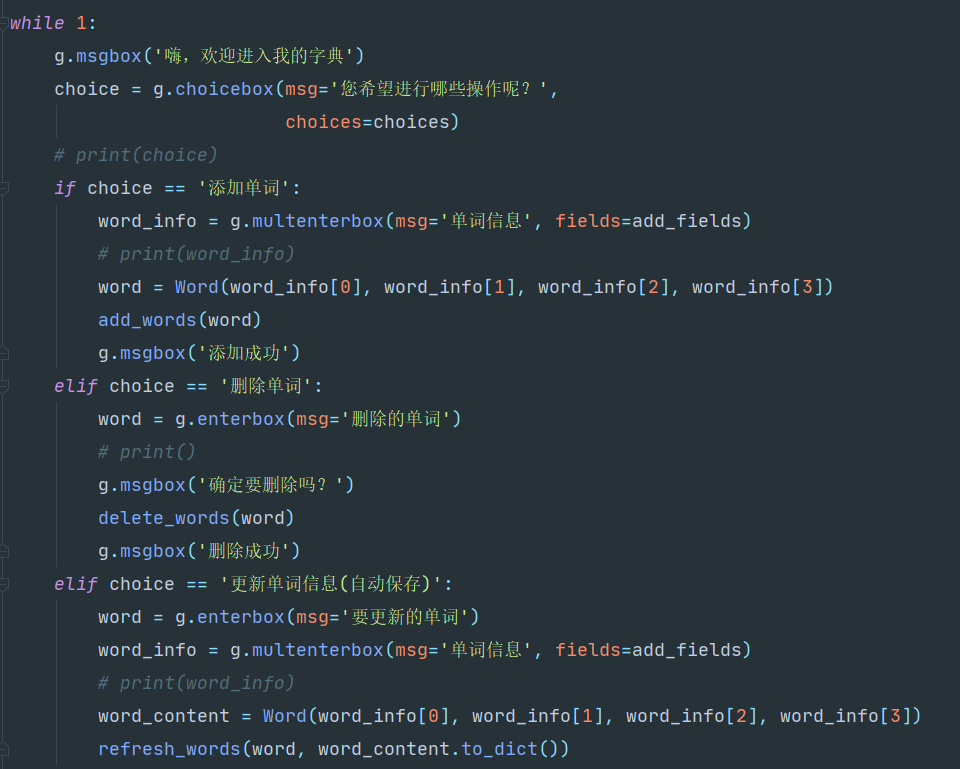


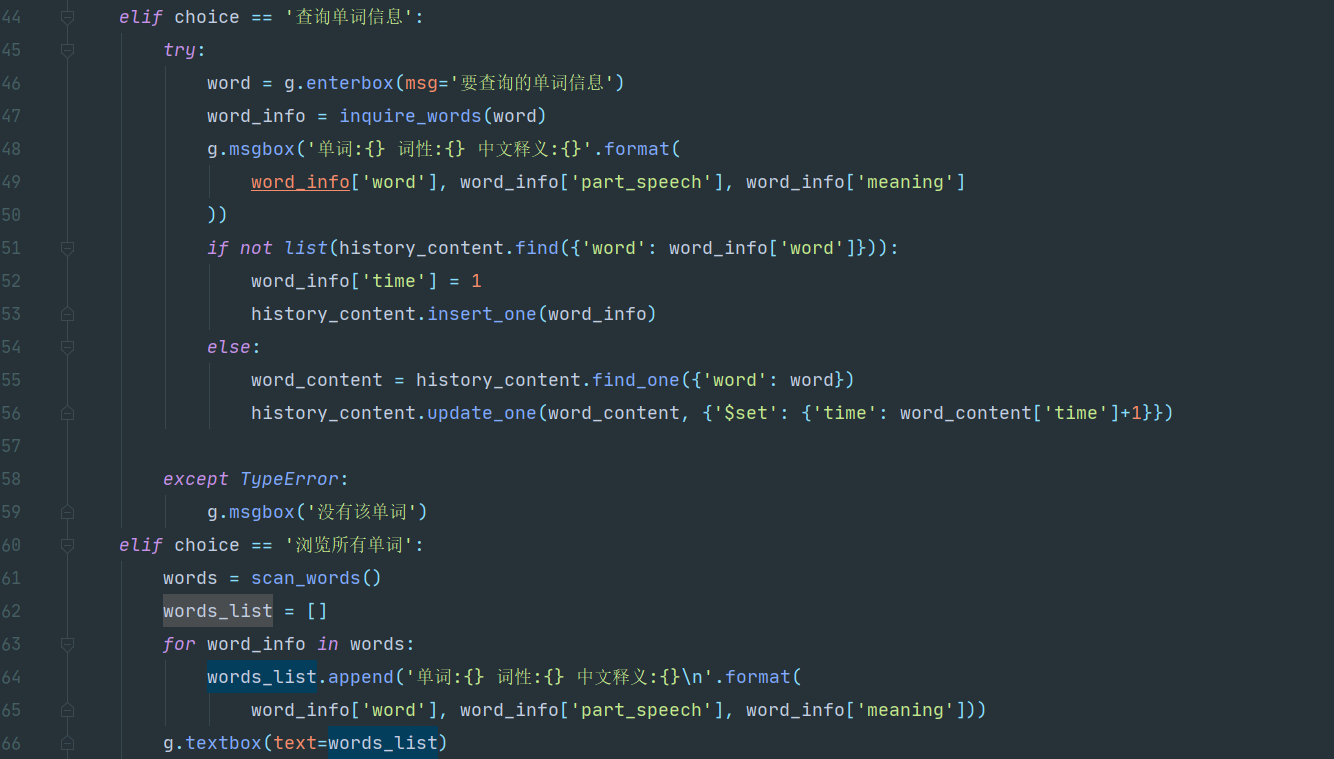
**Gui.py:**

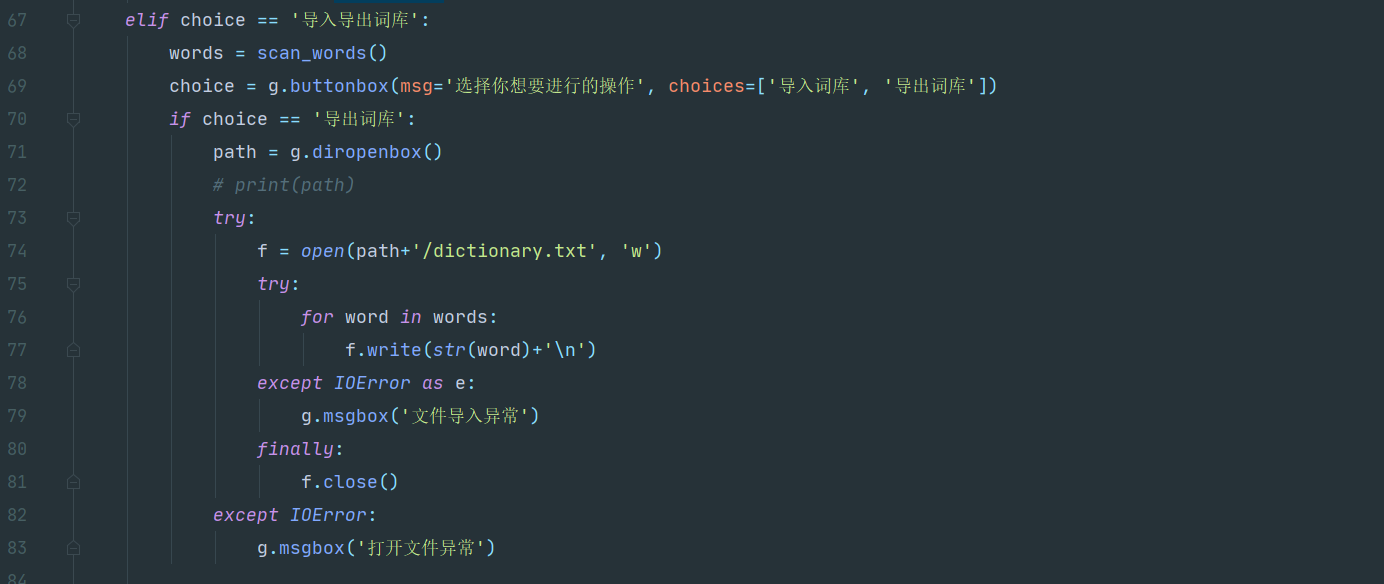
**定义各种属性以及列表**

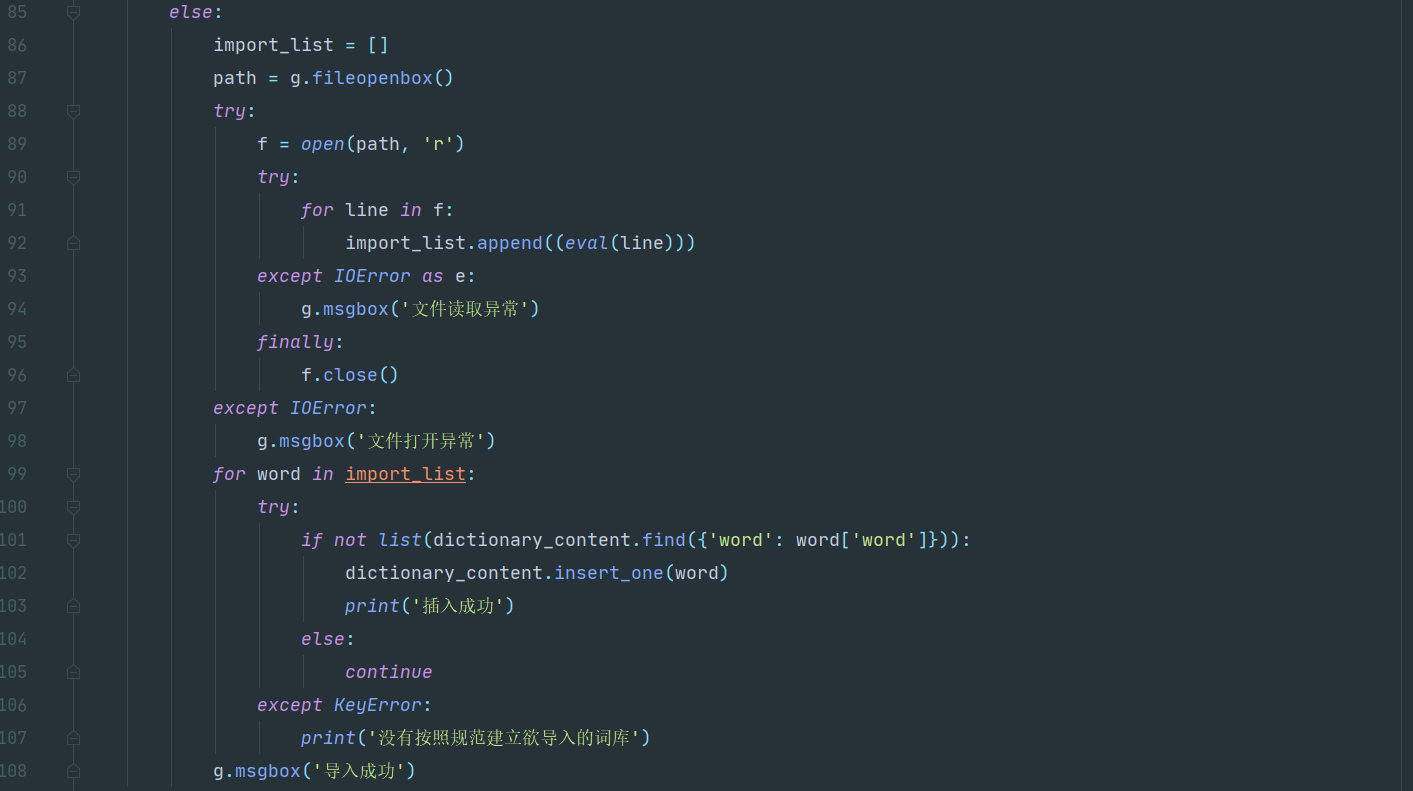


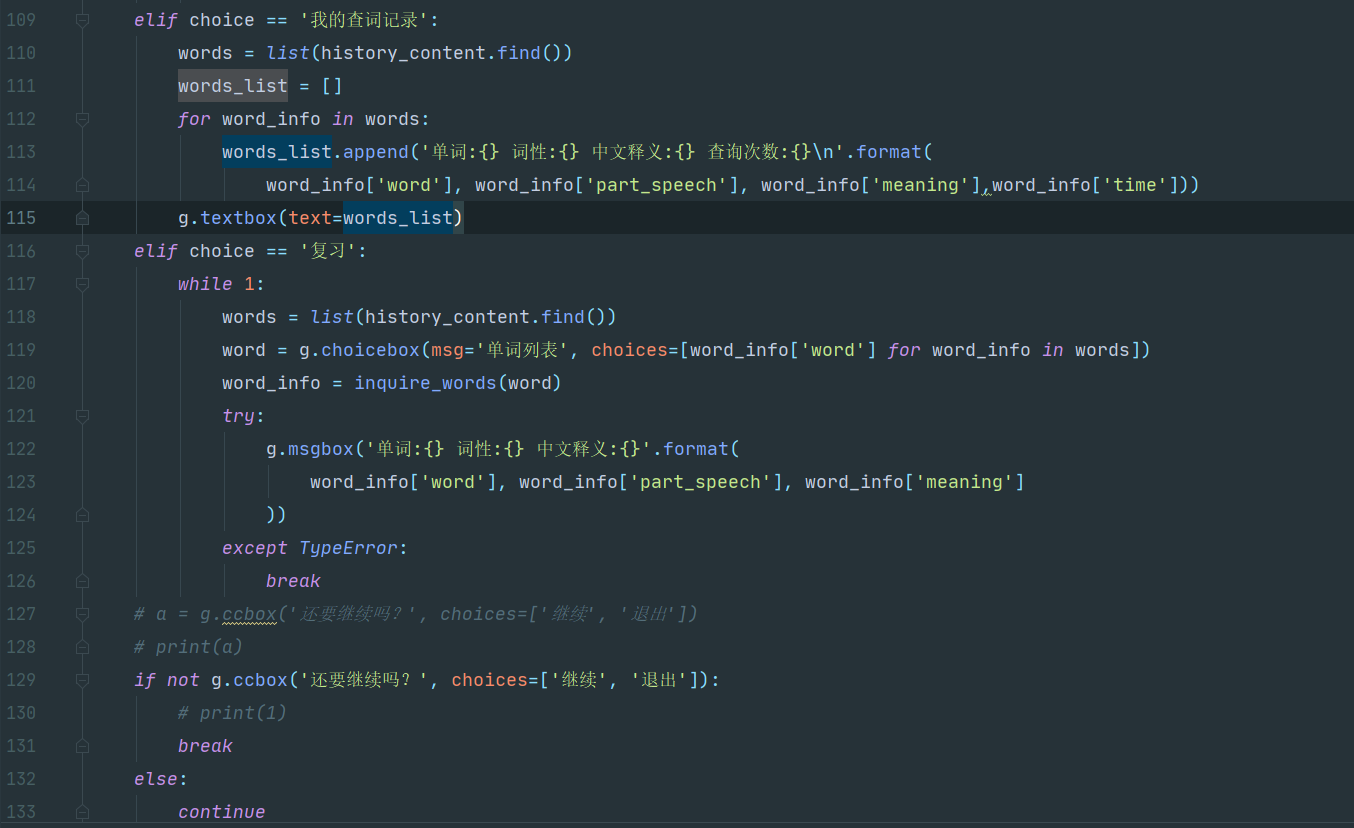
**主循环：**











**整个项目已经放在了我的github：**

<https://github.com/SQALRT/diction>