# 第1章 绪论

1.1 Python简介

Python是一种面向对象、直译式计算机程序设计语言，也是一种功能强大而完善的通用型语言，已经具有十多年的发展历史，成熟且稳定。这种语言具有非常简捷而清晰的语法特点，适合完成各种高层任务，几乎可以在所有的操作系统中运行。目前，基于这种语言的相关技术正在飞速的发展，用户数量急剧扩大，相关的资源非常多。

1.2网站：<http://www.5323391.com/>

1.3 网络爬虫的定义：

-模拟浏览器自动的浏览网页

-自动的批量的采集我们需要的网络资源

# 第2章 详细设计

2.1 程序总体设计

原理：

模拟浏览器http请求网站服务器，服务器对此进行http响应返回给浏览器，浏览器会根据响应的内容进行相应的解析，渲染后展示给用户

思路：

先进入网页指定小说的章节列表页，获取这一页面的所有章节列表信息，获取列表信息网页后，再对各个章节列表页进行详细爬取，通过标签元素和正则表达式爬取我们想要的小说内容，最后保存到txt文本文件中。

2.2 程序详细设计

技术：

①反爬虫技术：

（1）自定义请求头headers ——User-Agent

用Python的urlib标准库时，浏览器会识别到你是用Python访问，这样很容易被禁止访问。

headers={'User-Agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 BIDUBrowser/8.7 Safari/537.36'}

（2）导入time模块，设置time.sleep

设置时间间隔时间

Import time

time.sleep(2)

②正则表达式

#获取每一章的信息（标题、章节、url）  
div = re.findall(r'<div class="book\_list">.\*?</div>',html,re.S)[0] #匹配不可见字符 re.S  
chapter\_info\_list = re.findall(r'<a href="(.\*?)"(.\*?)</a>',div,re.S) #补上

③词频统计技术

def get\_words(txt):  
 fc = open('分词.txt', 'w', encoding='gbk')  
 seg\_list = jieba.cut(txt) #对文本进行分词  
 c = Counter()  
 for x in seg\_list:  
 if len(x) > 1 and x != '\r\n': #进行词频统计  
 c[x] += 1  
 print('常用词频度统计结果')  
 for (k, v) in c.most\_common(100): #遍历输出高频词  
 #print('%s%s %s %d' % (' ' \* (5 - len(k)), k, '' \* int(v / 3), v))  
 print('%s %d' % (k,v))  
 fc.writelines(str(c.most\_common(100)))  
 fc.close()

④词云

def get\_cloud():  
 hack\_mask = np.array(Image.open('./pkq.jpg'))  
 fd = open("全球精灵时代.txt", "r", encoding="gbk") # 打开文件  
 t = fd.read() # 读取文件，并存好  
 fd.close()  
 ls = jieba.lcut(t) # 对文本分词  
 txt = " ".join(ls) # 对文本进行标点空格化  
 w = wordcloud.WordCloud(font\_path="msyh.ttc", width=500, height=333, background\_color="white", mask=hack\_mask,max\_words=500, max\_font\_size=150) # 设置词云背景  
 w.generate(txt) # 生成词云  
 w.to\_file("wordcloud.png") # 保存词云图  
 fd.close()

⑤清洗数据

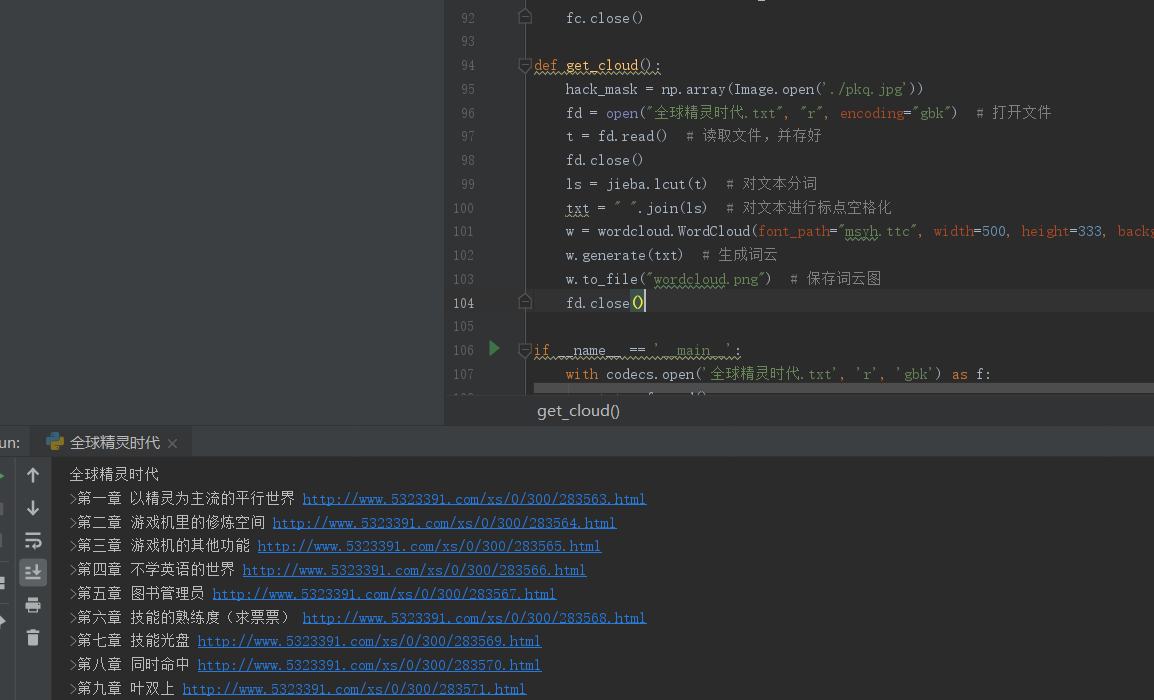
chapter\_content = chapter\_content.replace(' ','')  
chapter\_content = chapter\_content.replace('&nbsp;','')  
chapter\_content = chapter\_content.replace('<br>','')  
chapter\_content = chapter\_content.replace('<br/>','')

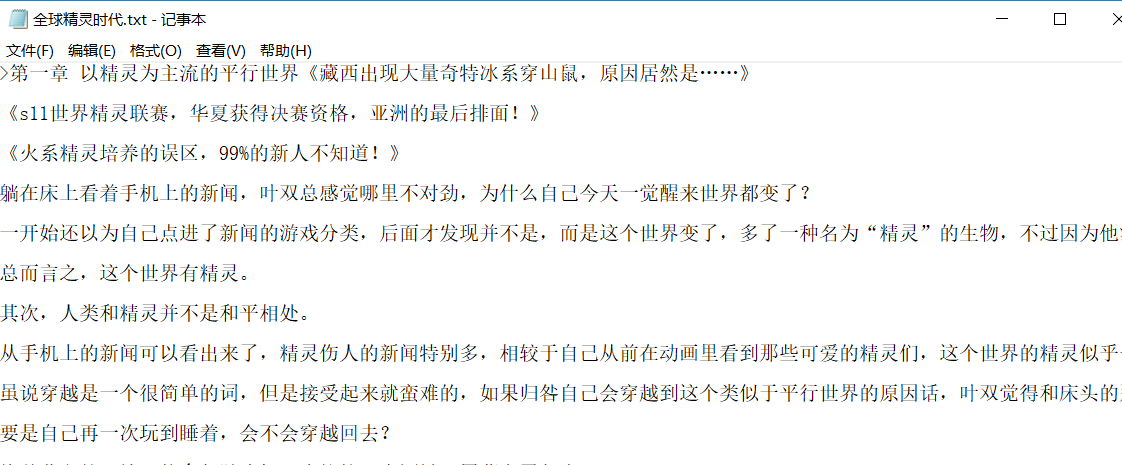
⑥保存文档

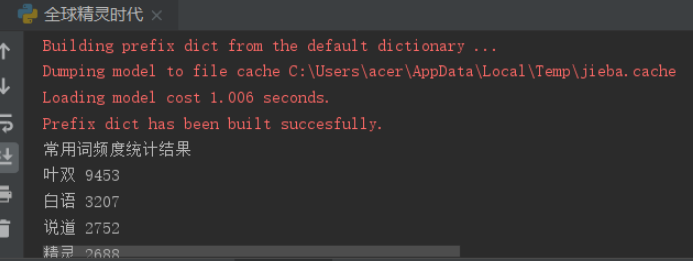
#新建一个文件，保存小说内容  
fb=open('%s.txt' %title,'w',encoding='gbk')

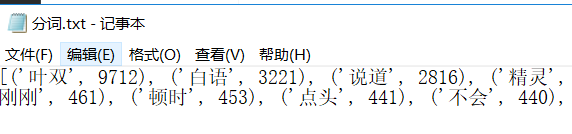
with codecs.open('全球精灵时代.txt', 'r', 'gbk') as f:  
 txt = f.read()

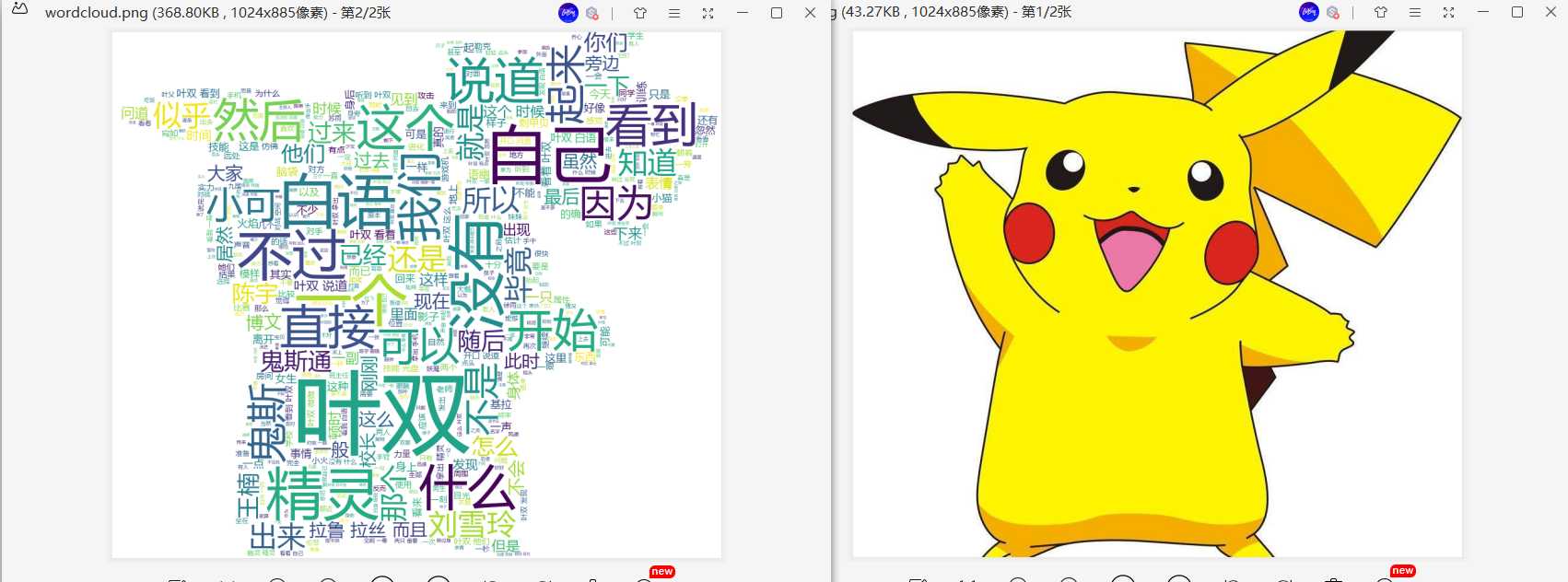
2.3 截图

爬取列表页的标题和url

小说保存为txt文本文件

对小说进行词频统计





词频统计结果保存为分词.txt并以词云形式展示

# 第3章 总结

通过两天的自主学习，掌握到了对图片，对小说的爬取，此次Python课程设计，让我巩固了不少Python的相关知识，培养了独立思考问题解决问题的能力，加深了对Python知识的理解，也做到了将所学知识加以运用,有助于今后的学习，这次课程设计也为自己积累很多宝贵的经验，而这些经验平时是很难得到的。