《创意编程课》学习复盘

清蒸 B18355

这个题目和名字学号一写吧，真的有种课程报告的感觉了啊哈哈。嗯，这本来就是一门很正经，很严肃的课。不过有些视频花絮也真的是精彩的不行，真希望以后的每个视频都留个彩蛋啊~~~看蜀黍的了。

说正题了，复习每次课的重点和要点。想了想这么有创造性的课，那我肯定不能按照第一次课到最后一次的顺序来总结啊。所以我决定，从最后一次到第一次开始复习，倒叙。我就打算直接按照自己的回忆写了，所以万一发生一点错乱，应该也不影响大局的啊哈哈。

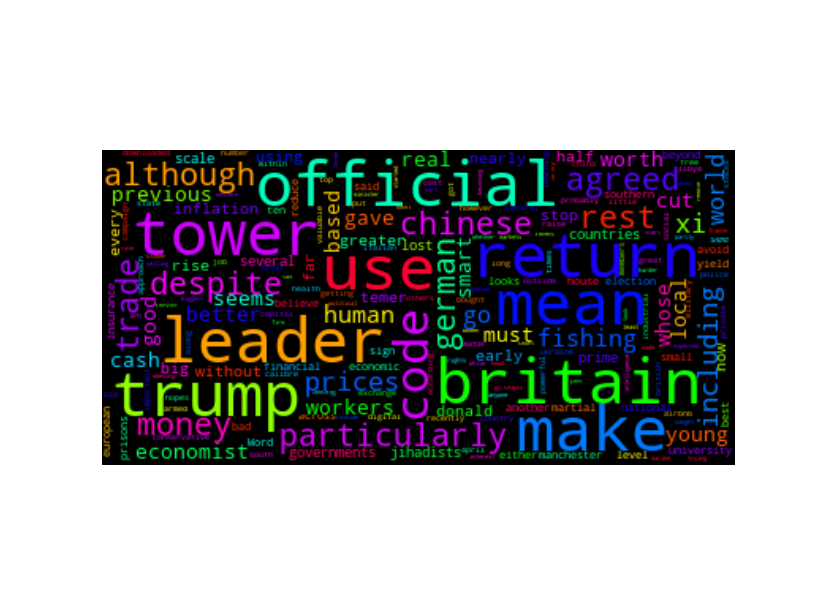
----------------------------------------------------------------------------------------------------------------

**1【关于大作业】**

----------------------------------------------------------------------------------------------------------------

1. 大作业是做一个荆棘雪人的热词分析，反正刚开始看到蜀黍在群里发的图云就觉得炒鸡炫酷啊。后来听说这就是大作业，真的开心，觉得自己也要炫酷起来了。
2. 【涉及到的一些知识】
3. **Class**

建立一个从指定路径中中读取.txt文件，并且可以进行一系列操作，最终将得到的统计数据传到wordcloud要用的文件中，输出图云文件。下面是我的一个看起来很像对的的图云



发现说了一堆好像并没有说这个类的事情，作为一个工科生，突然觉的列表说明可能好一些。

|  |  |  |
| --- | --- | --- |
| class WordReader | | |
| 方法 | 描述 | 用到主要知识点 |
| read\_file(self,path) | 从特定路径读入.txt文件内容到all\_text字符串 | 文件的三个函数：open(path),read(),close() |
| word\_split(self, all\_text) | 将读入的文本分割成单词存储在一个list中 | 用到了分割字符串的函数split()  将string的字母转化为小写的lower()函数  list添加元素的append()  list类型存储单词 |
| word\_count(self, list) | 遍历存储单词的list，计算得到每个单词的出现次数，存储在一个dict中，key/value对应word/count | list的遍历  字典dict类型存储word/count对  dict的has.key()查找函数 |
| word\_sort(self, dict) | 按照dict中值的大小进行排序，最终得到存储单词的列表key和存储相应次数的列表value | dict类型的keys()函数，取出所有的键值存为list  list自带的sort()排序方法 |
| write\_csv(path, key, value) | 将得到的key和value存入指定路径的.csv文件 | 写入.csv文件file(path,’w’)  writerows() |
| 在这个类的基础上，通过测试发现会有很多数字也被识别为单词，还有一些以特殊符号，比如$开始的，会影响结果，所以加了一个函数**redu(dict)**来做一些判断，排除掉这些不属于word的东西。 | | 主要用到strip()函数 |

其实，蜀黍的tips里面的方法才是真的方便，但是考虑是自己的复盘我就先写自己的这些吧。蜀黍的那些很需要再研究研究的。【其实，觉得很重要的就是要知道有很多资源是不用自己写，可以调用的，所以还是要见多才能识广，所以期待蜀黍教我们很多how to找这种库啊包啊的方法~】

----------------------------------------------------------------------------------------------------------------

**2【关于工具】**

----------------------------------------------------------------------------------------------------------------

工具的话前前后后也是有一些了，突然一下问我几个我还真答不上来，我一个一个数数吧，哈哈。（这个按照时间顺序数了哈）

1. 第一个当然就是**sublime**了，真的超方便，当然也很炫酷。后来发现html也可以用，matlab也可以，竟然latex也可以，酷到不行。比之前有的notepad++要好一些了！关键ctrl+B就可以运行代码了呀。棒！
2. 第二个就是命令行了吧，之前用Ubuntu有接触过一些，但因为现在电脑一般是用windows，所以这次我是都走地windows路线。原来windows有自带的powershell，也还好，但是讲道理，跟linux比起来，确实如蜀黍所说，很弱鸡。不过将就着用用也还好了，实在讲究不下去的时候，我就回宿舍打开笔记本上的Ubuntu来玩耍啦哈哈。
3. 第三个是git了，这个我之前也有了解，但是是理论了解了一下，实际没怎么用，然后到现在都忘差不多了。然后蜀黍讲这个的时候，就是讲怎么实际地去用，所以才真的有点理解到这个工具的重要性，或者说它存在的意义。而且那几个命令也简单啊，好用啊。这个真的是蜀黍教的真的有种醍醐灌顶的感觉了，很赞！
4. 第4个好像还推荐了一个sourcetree，和git放一起用的，但是我注册的时候总是不成功，所以我就放弃了。后来说要翻墙，表示不是很习惯翻墙，遂放弃。
5. 第5个呢，就是wordcloud了，如果也可以算进去的话。刚开始想着Ubuntu上好装一些，我就装在Ubuntu上了，到实验室之后，又在windows上安装，其实也不是很难嘛哈哈。所以就愉快地玩起来了，见第一幅图哈。其实这个地方我还是没有很认真的去看代码，对于这个的用法还是有点疑问的，当然蜀黍要是来个视频讲一发的话就很棒棒了！~

先这5个吧，其他的可能不是很重要，没记住，或者真的只有这五个哦。不还有在线的菜鸟教程，和一些线上运行代码的工具，都很赞！

----------------------------------------------------------------------------------------------------------------

**3【python基础语法】**

----------------------------------------------------------------------------------------------------------------

这里可能倒叙不了了，只能乱序了。

1. 两个原则
2. Single Responsibility Principle
3. Don’t Repeat Yourself

第一个SRP就是说，我举例说吧，比如要实现买菜和记账这个事情，写在一起也能实现，但是要考虑它们是不同的功能最好把他们分开写两个函数，各司其职。这样，下次买完书要记账就可以直接用记账的函数了呀~

第二个DRY原则当时也有醍醐灌顶的感觉，记得当时是两天的作业，然后前两天的对比之后，真的是恍然大悟。就是说，比如咱们要用到一个π值，都知道是约等于3.14，那如果你在代码里面每一个用到的地方都写了3.14，那么如果有天我觉得这个效果不行，不够精确，我要把π近似为3.1415927，那你是不是要一个地方一个地方找3.14这个数字了，知道还要判断它是不是你用的π，而不是表示其他含义的3.14，这就很烦了，不仅工作量大，还有可能漏改还可能错改。所以这种时候，DRY原则就出场了，定义PI=3.14，然后在代码中需要放π的地方都用PI代替，那改的时候，只需要改了PI=3.1415927就可以了，简单快捷不出错，何乐不为！

1. 三个数据结构

关于这三个的叙述我就算了吧，我写一点我自己遇到的一些容易出错的点。

1. 列表 list =[]
2. 列表元素下标从0开始
3. 列表元素可以用list[index]的方式访问，但是！！一定要注意，要index存在才可以访问，比如lst=[1,2,3]，也就是index=0,1,2。这个时候你要访问lst[3]就是你的不对了。我犯了很多次这种错误（估计是受MATLAB的影响），每次想把一些东西存到一个空列表时，我就会直接这样访问，实际是不行的。这种情况应该用append()函数，把要存的东西通过这个函数加到list后面。
4. list还有自带的很多功能可以用，比如，in, not in, insert, del, sort等等。
5. 字典 dict = {}
6. 字典是key/value对为元素组成的数据结构。
7. key和value之间的连接方式：key:value, 我也不知道我总是用等号，一定是冒号啊！！
8. 字典的一些自带功能，has.key()查找是否存在某键，keys()返回所有键为一个list
9. 字典的访问：dict[key] = value
10. 字典的添加：dict[new\_key] = new\_value 【和list的区别啊】
11. 元组

元组理解就是不可以进行修改的list，书上说它是不可变的。举例子呢，就是比如性别这个选项，【此处只考虑普通两种】，只能有男和女啊，就算你有权限改成狗这种选项，很不合适啊，所以存储这种类似属性，就用到元组啦。

对，补充一个list和元组共同的切片相关

1. 首先list1=list，这种情况下，list1只是list的一个引用，也就是说，对list1的修改直接对list产生影响；那我们只是单纯的想复制的时候怎么办呢，借助切片：list1=list[:], 这样就可以了，这样就会在一个新的地方存储list1，对list1的修改不再对list有影响。
2. 切片要注意前后，具体例子太多，可以考虑网上查阅，我就不写啦。

----------------------------------------------------------------------------------------------------------------

4**【python基础语法-2】**

----------------------------------------------------------------------------------------------------------------

这个地方就是一些变量的定义复制，以及形参和实参的传递之类的问题了，因为之前有点编程负基础，所以这个地方基本都是理解的。没有什么理解问题。所以印象也不是很深，印象比较深的是蜀黍老妈买菜这个连续的故事啦哈哈哈。用一个故事来学编程很有一种在看小说的感觉了。

----------------------------------------------------------------------------------------------------------------

5**【python关于库】**

----------------------------------------------------------------------------------------------------------------

1. 最好玩的库当然还是turtle啦，真心不错（工科生只能写成这样了），从这个好玩的出发，算是对python的库有了一个很基础的了解，蜀黍也分享了查询的方式，如何使用啊之类的。
2. 后来又自己去联系了一些库，比如time，datetime还有calendar这些库，但是问题就是练了之后也看懂了，是觉得很厉害，但感觉自己没有用到，也不知道在哪里用。但貌似这几个库又是非常有用的那几个，所以还是希望有一些有用或者有意思的作业可以来练习一下，加深一下印象，可以让我知道什么时候该去想到用这些库。

----------------------------------------------------------------------------------------------------------------

**6【关于拓展学习】**

----------------------------------------------------------------------------------------------------------------

1. 《python简明教程》

这本书的电子版是在前几次课就提供出来给我们参考的，当时对python属于一无所知的状态。在刚开始几课倒也还行，但是同步阅读这本书，做一些上面的例子，对后面的学习是非常有用的。比较像正式学习之前的预习，也像是上课讲了一些重要的东西，还有一些类似的东西在这书上可以进一步扩展。所以，真的建议初学者把这本书读一下，做一下上面的例子。

1. 菜鸟教程

这个教程我之前就知道的，现在网络上资源其实很多，有时候自己课余时间去查阅一下，翻看一下，理解一下不同的例子，是可以对一个概念有更深的理解的。我后来开始做菜鸟教程上面的100道练习题，现在超过50道了吧，自己先做，然后再去网上对答案，对比一下，收货很大。而且一般网上的答案不止一种，可以了解很多殊途同归的方法，这样学的话觉得很快，也很有成就感。

1. 一些网络资源

比如说蜀黍给的一些可以查阅python库的地方，之前是编程负基础嘛，所以基础也就只是知道学这东西考大学计算机然后结束了，能算个题目啥的，写个加减乘除。库才让我知道原来有很多现成的很厉害的东西可以用啊！