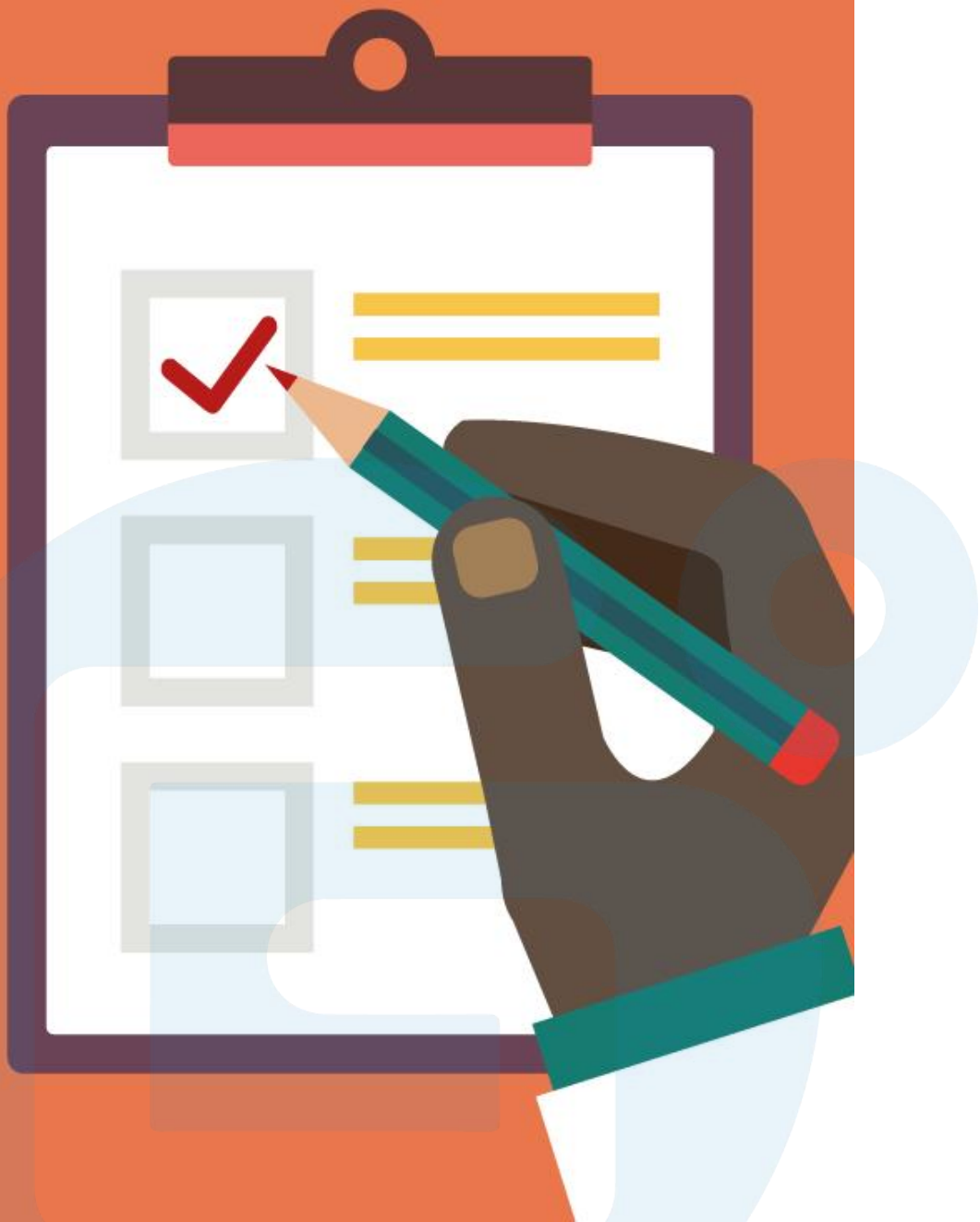


Talk is cheap, show me the code

第八课：Python词频统计

Python初阶入门课程系列



OUTLINE

➤ 文件操作

➤ 词频统计



一 文件操作



什么是文件？

- 文件是数据的抽象和集合
 - 文件是存储在外置存储器上的数据序列
 - 文件是数据存储的一种形式
 - 文件展现形态：文本文件和二进制文件
-

文本文件和二进制文件？

- 文件文件和二进制文件只是文件的展示方式，本质上，所有文件都是二进制形式存储。形式上，所有文件采用两种方式展示。
- **文本文件**由单一特定编码组成的文件，如UTF-8编码。由于存在编码，也被看成是存储着的长字符串。适用于例如：.txt文件、.py文件等。
- **二进制文件**直接由比特0和1组成，没有统一字符编码。一般存在二进制0和1的组织结构，即文件格式。适用于例如：.png文件、.avi文件等。

文件操作

```
1  ###1
2  a="寒雨连江夜入吴,平明送客楚山孤"
3  print(a)
4  print(a.encode()) #默认是utf-8编码, 中文3个字节
5  ###2
6  f = open("f.txt", "w")
7  f.writelines(a)
8  f.close()
9  ###3
10 tf = open("f.txt", "rt")
11 print(tf.readline())
12 tf.close()
13 ###4
14 bf = open("f.txt", "rb") #windows默认gbk编码 (ANSI编码), 中文2字节
15 print(bf.readline())
16 bf.close()
17
18
```

控制台 1/A

```
In [11]: runfile('C:/Users/Isaac/Desktop/test/test1.py', wdir='C:/Users/Isaac/Desktop/test')
```

寒雨连江夜入吴,平明送客楚山孤

```
b'\xe5\xaf\x92\xe9\x9b\xa8\xe8\xbf\x9e\xe6\xb1\x9f\xe5\xa4\x9c\xe5\x85\xa5\xe5\x90\xb4,\xe5\xb9\xb3\xe6\x98\x8e\xe9\x80\x81\xe5\xae\xa2\xe6\xa5\x9a\xe5\xb1\xb1\xe5\xad\xa4'
```

寒雨连江夜入吴,平明送客楚山孤

```
b'\xba\xae\xd3\xea\xc1\xac\xbd\xad\xd2\xb9\xc8\xeb\xce\xe2,\xc6\xbd\xc3\xf7\xcb\xcd\xbf\xcd\xb3\xfe\xc9\xbd\xb9\xc2'
```

文件操作

- 文件处理的步骤: 打开-操作-关闭
- 操作包括读取, 写入等。
- 打开的本质是将文件从外置的存储器取出数据, 放置入内存空间以便操作。
- 关闭则相反。

打开:

<变量名> = open(<文件名>, <打开模式>)

文本 or 二进制
读 or 写

文件句柄

文件路径和名称
如果和源文件同目录可省掉路径

文件打开模式	描述
'r'	只读模式，默认值，如果文件不存在，返回FileNotFoundError
'w'	覆盖写模式，文件不存在则创建，存在则完全覆盖
'x'	创建写模式，文件不存在则创建，存在则返回FileExistsError
'a'	追加写模式，文件不存在则创建，存在则在文件最后追加内容
'b'	二进制文件模式
't'	文本文件模式，默认值
'+'	与r/w/x/a一同使用，在原功能基础上增加同时读写功能


```
f = open("f.txt")
```

```
f = open("f.txt", "rt")
```

```
f = open("f.txt", "w")
```

```
f = open("f.txt", "a+")
```

```
f = open("f.txt", "x")
```

```
f = open("f.txt", "b")
```

```
f = open("f.txt", "wb")
```

- 文本形式、只读模式、默认值
- 文本形式、只读模式、同默认值
- 文本形式、覆盖写模式
- 文本形式、追加写模式+ 读文件
- 文本形式、创建写模式
- 二进制形式、只读模式
- 二进制形式、覆盖写模式

关闭:

<句柄名>.close()

打开以后的文件读取

操作方法	描述
<code><f>.read(size=-1)</code>	读入全部内容，如果给出参数，读入前size长度 >>> s = f.read(2)
<code><f>.readline(size=-1)</code>	读入一行内容，如果给出参数，读入该行前size长度 >>> s = f.readline()
<code><f>.readlines(hint=-1)</code>	读入文件所有行，以每行为元素形成列表 如果给出参数，读入前hint行 >>> s = f.readlines()

打开以后的文件读取和写入

- 文件内容的读取: `.read()` `.readline()` `.readlines()`
- 数据的文件写入: `.write()` `.writelines()` `.seek()`



二 词频统计



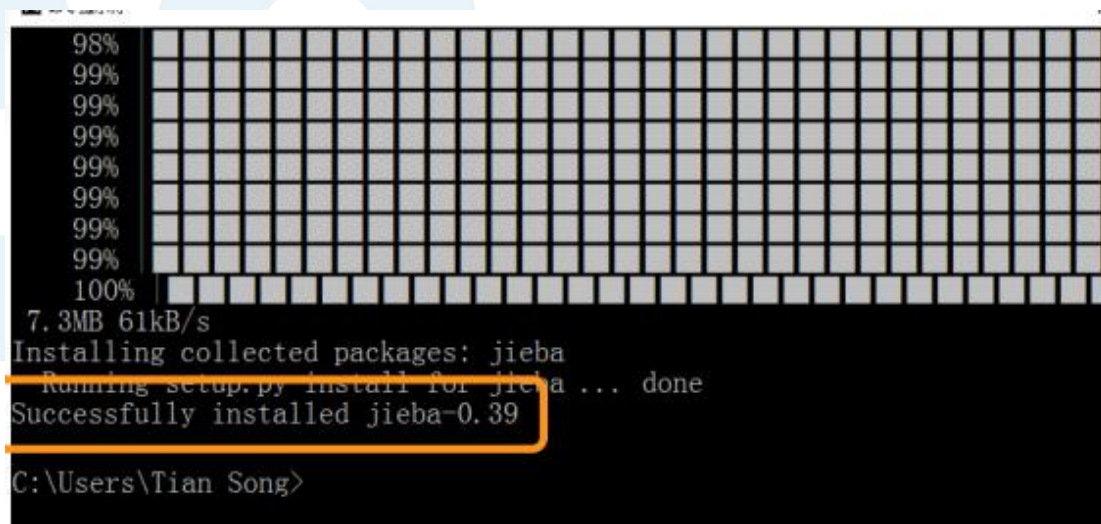
词频统计

Anaconda以外的第三方库需要两个。

jieba库和wordcloud库，分别用于中文的分词和词云显示

其余均为自带库

anaconda prompt里 pip install jieba和pip install wordcloud即可。

A screenshot of a Windows command prompt window. The window shows the progress of installing the 'jieba' package. At the top, there is a progress bar with a grid of squares, mostly filled, indicating 100% completion. Below the progress bar, the text '7.3MB 61kB/s' is displayed. The main text in the terminal reads: 'Installing collected packages: jieba', 'Running setup.py install for jieba ... done', and 'Successfully installed jieba-0.39'. The prompt 'C:\Users\Tian Song>' is visible at the bottom.

```
98%
99%
99%
99%
99%
99%
99%
99%
99%
100%
7.3MB 61kB/s
Installing collected packages: jieba
Running setup.py install for jieba ... done
Successfully installed jieba-0.39
C:\Users\Tian Song>
```

制作词云图

词云图制作前，需要先准备几个东西：

- 下载python wordcloud库，也是词图库制作的关键库；
- 如果要对中文句子进行分词，那么需要jieba库；如果是英文分词，那可以不下载；

为什么要分词？

“中华 人民 共和国 在 1949年10月1日 成立了！”

原理：隐马尔可夫模型(HMM)

- 利用一个中文词库，确定中文字符之间的关联概率
 - 中文字符间概率大的组成词组，形成分词结果
 - 除了分词，用户还可以添加自定义的词组
-

三种模式

精确模式：把文本精确的切分开，不存在冗余单词

全模式：把文本中所有可能的词语都扫描出来，有冗余

搜索引擎模式：在精确模式基础上，对长词再次切分

```
1 import jieba
2 a="中华人民共和国在1949年10月1日成立了！"
3 print(jieba.lcut(a)) #lcut 精确模式
4
5 print(jieba.lcut(a,cut_all=True)) #lcut 全模式
6
7 print(jieba.lcut_for_search(a)) #搜索引擎模式
```

控制台 1/A

```
In [20]: runfile('C:/Users/Isaac/Desktop/test/test1.py', wdir='C:/Users/Isaac/Desktop/test')
['中华人民共和国', '在', '1949', '年', '10', '月', '1', '日', '成立', '了', '！']
['中华', '中华人民', '中华人民共和国', '华人', '人民', '人民共和国', '共和', '共和国', '在', '1949', '年', '10', '月', '1', '日', '成立', '了', '！']
['中华', '华人', '人民', '共和', '共和国', '中华人民共和国', '在', '1949', '年', '10', '月', '1', '日', '成立', '了', '！']
```


词频统计

```
1  # -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4
5  This is a temporary script file.
6  """
7  import jieba
8  txt = open("分析文档.txt", encoding="utf-8").read()
9  #加载停用词表
10 stopwords = [line.strip() for line in open("停用词库.txt", encoding="utf-8").readlines()]
11 words = jieba.lcut(txt)
12 counts = {}
13 for word in words:
14     #不在停用词表中
15     if word not in stopwords:
16         #不统计字数为一的词
17         if len(word) == 1:
18             continue
19         else:
20             counts[word] = counts.get(word,0) + 1
21 items = list(counts.items())
22 items.sort(key=lambda x:x[1], reverse=True)
23 for i in range(30):
24     word, count = items[i]
25     print("{:<10}{:>7}".format(word, count))
```

In [21]: runfile('G:/:

一个	512
翠翠	326
祖父	302
地方	183
事情	169
母亲	163
船夫	149
二老	144
明白	128
萧萧	109
一种	103
家中	103
一点	94
碾坊	92
渡船	86
东西	85
自然	85
爷爷	84
声音	84
有人	82
船上	77
城里	77
两人	72
唱歌	70
水手	70
过渡	60

利用字典表达词频

利用词云直观的表达文本频率



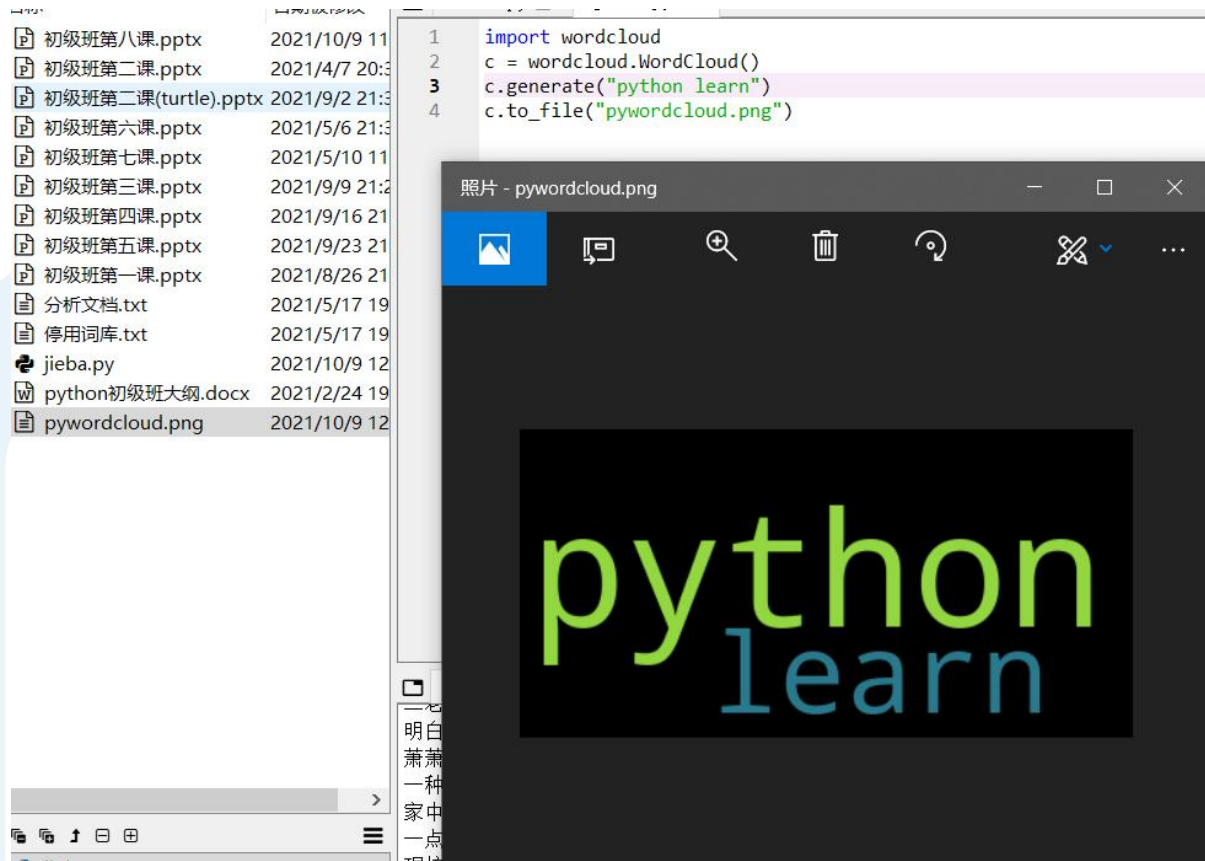
利用词云直观的表达文本频率

- wordcloud库把词云当作一个WordCloud对象，wordcloud.WordCloud()代表一个文本对应的词云，

```
w = wordcloud.WordCloud()
```

- 可以根据文本中词语出现的频率等参数绘制词云
- 词云的绘制形状、尺寸和颜色都可以设定，以WordCloud对象为基础配置参数、加载文本、输出文件。

利用词云直观的表达文本频率



- ① 分隔: 以空格分隔单词
- ② 统计: 单词出现次数并过滤
- ③ 字体: 根据统计配置字号
- ④ 布局: 颜色环境尺寸


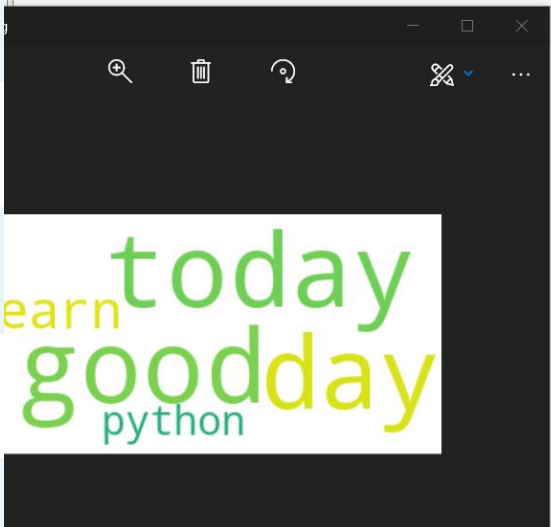
w = wordcloud.WordCloud(<参数>)

参数	描述
width	指定词云对象生成图片的宽度，默认400像素 <code>>>>w=wordcloud.WordCloud(width=600)</code>
height	指定词云对象生成图片的高度，默认200像素 <code>>>>w=wordcloud.WordCloud(height=400)</code>
min_font_size	指定词云中字体的最小字号，默认4号 <code>>>>w=wordcloud.WordCloud(min_font_size=10)</code>
max_font_size	指定词云中字体的最大字号，根据高度自动调节 <code>>>>w=wordcloud.WordCloud(max_font_size=20)</code>
font_step	指定词云中字体字号的步进间隔，默认为1 <code>>>>w=wordcloud.WordCloud(font_step=2)</code>
font_path	指定字体文件的路径，默认None <code>>>>w=wordcloud.WordCloud(font_path="msyh.ttc")</code>
max_words	指定词云显示的最大单词数量，默认200 <code>>>>w=wordcloud.WordCloud(max_words=20)</code>
stop_words	指定词云的排除词列表，即不显示的单词列表 <code>>>>w=wordcloud.WordCloud(stop_words={"Python"})</code>
mask	指定词云形状，默认为长方形，需要引用imread()函数 <code>>>>from scipy.misc import imread >>>mk=imread("pic.png") >>>w=wordcloud.WordCloud(mask=mk)</code>
background_color	指定词云图片的背景颜色，默认为黑色 <code>>>>w=wordcloud.WordCloud(background_color="white")</code>

词频统计

```
1 import jieba
2 import wordcloud
3
4 ###1
5 txt = "today is a good day to learn python"
6 w = wordcloud.WordCloud(background_color = "white")
7 w.generate(txt)
8 w.to_file("pywcloud1.png")
9
10 ###2
11 txt = "过去一年，在新中国历史上极不平凡。面对突如其来的新冠肺炎疫情、\
12 世界经济深度衰退等多重严重冲击，在以习近平同志为核心的党中央坚强领导下，\
13 全国各族人民顽强拼搏，疫情防控取得重大战略成果，\
14 在全球主要经济体中唯一实现经济正增长，脱贫攻坚战取得全面胜利\
15 ，决胜全面建成小康社会取得决定性成就，\
16 交出一份人民满意、世界瞩目、可以载入史册的答卷。\\
17 全年发展主要目标任务较好完成，\\
18 我国改革开放和社会主义现代化建设又取得新的重大进展。"
19
20 w = wordcloud.WordCloud( width=1000,\
21 font_path="msyh.ttf",height=700)
22 w.generate(" ".join(jieba.lcut(txt)))
23 w.to_file("pywcloud2.png")
```

控制台 1/A





感谢参与 下堂课见

