

2022 年 12 月一实验七

【实验名称】

数据及文件操作练习

【实验目的】

1. 统计 hamlet.txt 中每个单词的出现频次，输出频次最高的 10 个单词。
 - 1) 保留介词、冠词、连词的情况下统计词频，并输出结果
 - 2) 去除介词、冠词、连词的情况下统计词频，并输出结果
2. 将整数 12345 分别写入文本文件 test.txt 和二进制文件 test.dat，并比较两个文件的不同输出。

【实验内容】

1、程序清单

(1) 统计每个单词的出现频次

1) 保留介词、冠词、连词的情况下统计词频

```
import re

res = {}
# 只读打开文件
with open('./hamlet.txt', 'r') as f:
    txt = f.read()

for line in txt.splitlines():
    line = re.sub(r'[+=$#!]', ' ', line) # 去除所有标点符号
    for word in line.split():
        flag = False
        if word[-1] == '-':
            up = word[:-1]
            flag = True
            break
        if flag:
            word = up + word # 拼接末位单词
            flag = False
        res.setdefault(word.lower(), 0)
        res[word.lower()] += 1
values = sorted(res.values())

sortedres = sorted(res.items(), key=lambda d: d[1], reverse=True)
for i in range(0, 10):
    print(sortedres[i][0])
```

2) 去除介词、冠词、连词的情况下统计词频

```
import re

res = {}
# 只读打开文件
with open('./hamlet.txt', 'r') as f:
    txt = f.read()

for line in txt.splitlines():
    line = re.sub(r'[+=$#!]', ' ', line) # 去除所有标点符号
    for word in line.split():
        flag = False
        if word[-1] == '-':
            up = word[:-1]
            flag = True
            break
        if flag:
            word = up + word # 拼接末位单词
            flag = False
        res.setdefault(word.lower(), 0)
        res[word.lower()] += 1
values = sorted(res.values())

sortedres = sorted(res.items(), key=lambda d: d[1], reverse=True)

lista = ['the', 'a', 'an', 'at', 'on', 'behind', 'during', 'from',
        'into', 'and', 'but', 'or', 'so', 'however', 'although']
count = 0
for i in range(0, 20):
    if sortedres[i][0] not in lista:
        print(sortedres[i][0])
        count += 1
    if count == 10:
        break
```

(2) 将整数 12345 分别写入文本文件 test.txt 和二进制文件 test.dat

```
a = '12345'

with open('./test.txt', 'w') as f:
    f.write(a)

with open('./test.dat', 'w') as f:
    f.write(a)
```

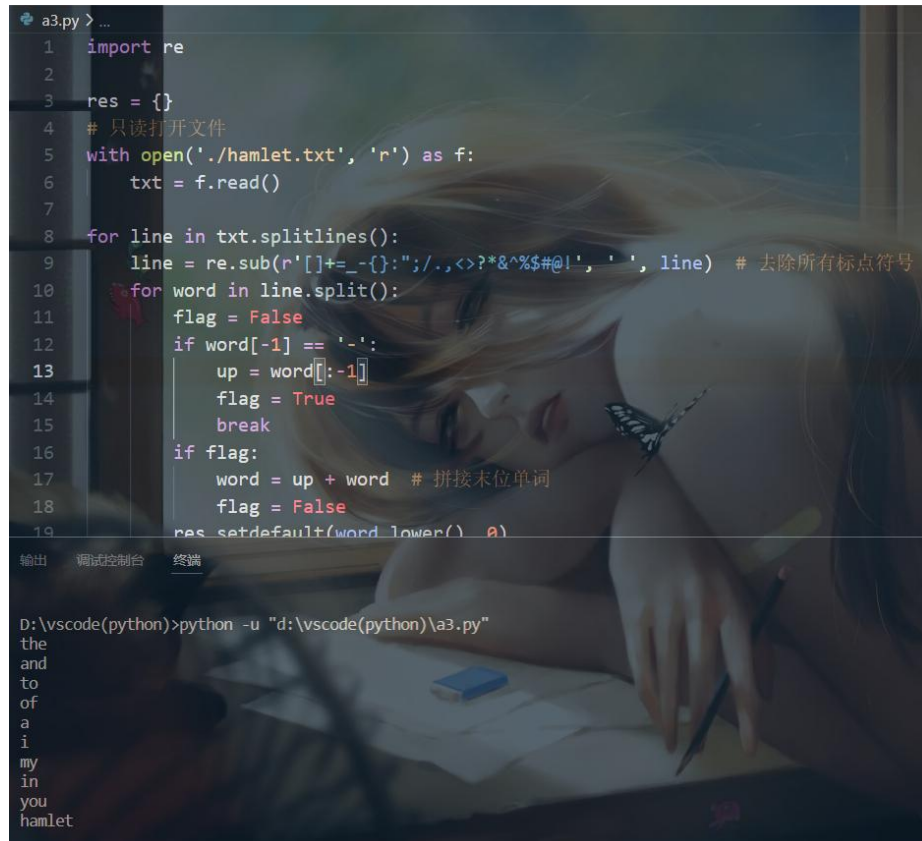
```
with open('./test.txt', 'r') as f:
    print(f.read())
```

```
with open('./test.dat', 'r') as f:
    print(f.read())
```

2、结果截图

(1) 统计每个单词的出现频次

1) 保留介词、冠词、连词的情况下统计词频



```
a3.py > ...
1  import re
2
3  res = {}
4  # 只读打开文件
5  with open('./hamlet.txt', 'r') as f:
6      txt = f.read()
7
8  for line in txt.splitlines():
9      line = re.sub(r'[\s+_-{}:":/.,<>?*%^$#@!]', ' ', line) # 去除所有标点符号
10     for word in line.split():
11         flag = False
12         if word[-1] == '-':
13             up = word[:-1]
14             flag = True
15             break
16         if flag:
17             word = up + word # 拼接末位单词
18             flag = False
19     res.setdefault(word.lower(), 0)

输出 调试控制台 终端

D:\vscode(python)>python -u "d:\vscode(python)\a3.py"
the
and
to
of
a
i
my
in
you
hamlet
```

2) 去除介词、冠词、连词的情况下统计词频

```
a3.py > ...
1  import re
2
3  res = {}
4  # 只读打开文件
5  with open('./hamlet.txt', 'r') as f:
6      txt = f.read()
7
8  for line in txt.splitlines():
9      line (variable) word: str ' ', line) # 去除所有标点符号
10     for word in line.split():
11         flag = False
12         if word[-1] == '-':
13             up = word[:-1]
14             flag = True
15             break
16         if flag:
17             word = up + word # 拼接末位单词
18             flag = False
19         res.setdefault(word.lower(), 0)
20         res[word.lower()] += 1
21     values = sorted(res.values())
```

输出 调试控制台 终端

```
D:\vscode(python)>python -u "d:\vscode(python)\a3.py"
to
of
i
my
in
hamlet
you
that
it
is
```

(2) 将整数 12345 分别写入文本文件 test.txt 和二进制文件 test.dat

```
1  a = '12345'
2
3  with open('./test.txt', 'w') as f:
4      f.write(a)
5
6  with open('./test.dat', 'w') as f:
7      f.write(a)
8
9  with open('./test.txt', 'r') as f:
10     print(f.read())
11
12 with open('./test.dat', 'r') as f:
13     print(f.read())
14
```

输出 调试控制台 终端

```
D:\vscode(python)>python -u "d:\vscode(python)\11.3(1).py"
12345
12345
```

【实验体会】

通过此次实验，使我更加熟悉了 `python` 中对于文件的操作和对于数据的处理，在数据的处理中，回顾了 `re` 库——正则表达式的使用，收获较大。