程设第七次作业

20377383 樊思涵

将本次作业分为四个部分，分别于 4 个’.py’文件实现

Task1

调用库

```python
import os
from functools import wraps
```

待创建文件夹的路径

```python
path = r'C:\Users\LF\Desktop\week8_task1'
```

被修饰的原始函数

```python
@check_path
def save(path,name_txt,txt):
    filenme = path + '\\' + name_txt + '.txt'
    with open(filenme,"w") as f:
        f.write(txt)
```

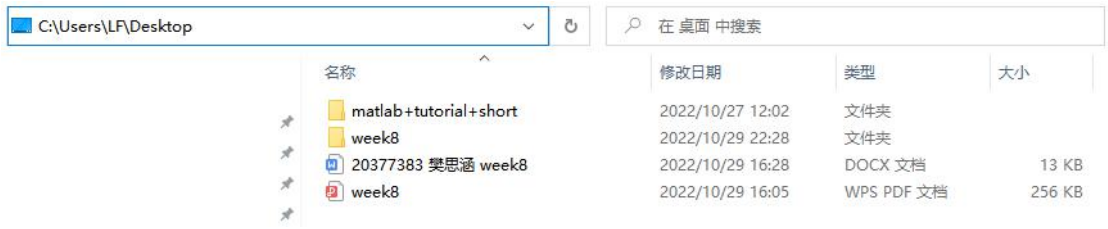
功能：在指定目录保存文本，并规定文件名

修饰器函数

```python
def check_path(func):
    @wraps(func)
    def wrapper(*args,**kwargs):
        print("path is "+args[0])
        if not os.path.exists(args[0]):
            print("There's no such path.New is created.\n")
            os.mkdir(path)
        else:
            print("Path exists.\n")
        return func(*args,**kwargs)
    return wrapper
```

结果展示：

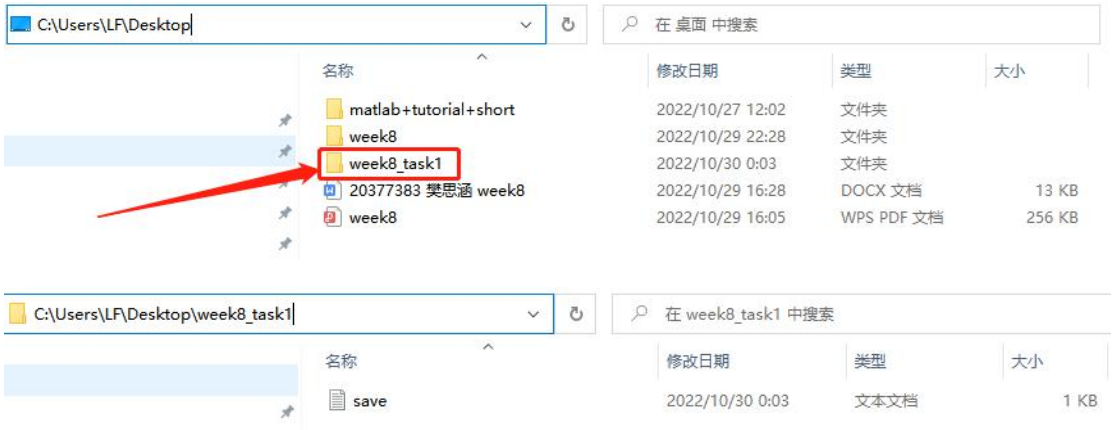
初始文件地址无目标文件夹



输出结果

```
PS E:\code\py_code> python -u "e:\code\py_code\week8\week8_task1.py"
path is C:\Users\LF\Desktop\week8_task1
There's no such path.New is created.
```

运行程序后目标文件夹成功建立并保存文件



已有目标文件夹后运行结果

```
PS E:\code\py_code> python -u "e:\code\py_code\week8\week8_task1.py"
path is C:\Users\LF\Desktop\week8_task1
Path exists.
```

Task2

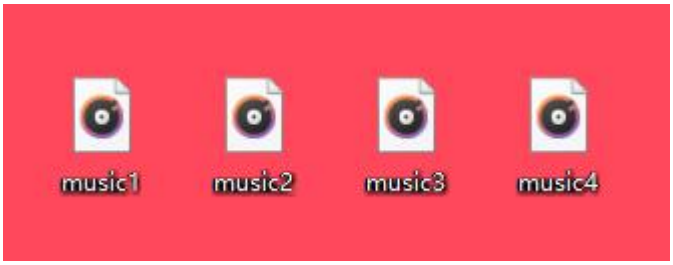
调用库

```
1  from functools import wraps
2  from playsound import playsound
3  from time import sleep
```

音乐文件路径

```
filename1 = r"C:\Users\LF\Desktop\music1.mp3"
filename2 = r"C:\Users\LF\Desktop\music2.mp3"
filename3 = r"C:\Users\LF\Desktop\music3.mp3"
filename4 = r"C:\Users\LF\Desktop\music4.mp3"
```

本地音乐文件



实现装饰器类

```
class Remind:
    """
    修饰器类
    """
    def __init__(self):
        pass

    def __remind(self,type_res):
        """
        内部方法
        用作实现修饰器中的提醒功能
        """
        if type_res == int:
            playsound(filename1)
        elif type_res == str:
            playsound(filename2)
        elif type_res in [list,dict,tuple]:
            playsound(filename3)
        else:
            playsound(filename4)

    def __call__(self,func):
        @wraps(func)
        def wrapper(*args,**kwargs):
            res = func(*args,**kwargs)
            for i in range(len(res)):
                self.__remind(type(res[i]))
            return res
        return wrapper

```

被修饰的测试用函数

```
40  @Remind()
41  def fun_test(*args,**kwargs):
42      """
43      用作测试
44      返回输入的参数
45      """
46      sleep(5)
47      return *args,*kwargs
48
```

在主函数中调用

```
49  def main():
50      a = 1
51      b = [1,2]
52      c = {'a':1,'b':2}
53      print(fun_test(b,a,c))
54
55  if __name__ == '__main__':
56      main()
```

发现音乐可以按指定顺序播放

Task3

调用库

```
1  from functools import wraps
2  import sys
```

保存输出文件目录

```
4  path = r'C:\Users\LF\Desktop'
```

带参数的修饰器

```python
def saveprint(path):
    """
    带参数的修饰器
    """
    def decorator(func):
        @wraps(func)
        def wrapper(*args,**kwargs):
            sys.stdout = open(path + '\\print.log', mode = 'w',encoding = 'utf-8')
            res = func(*args,**kwargs)
            return res
        return wrapper
    return decorator

```

被修饰的测试用函数

```python
@saveprint(path)
def fun_test():
    """
    测试用函数
    """
    for i in range(5):
        print(i)
```

在主函数中调用

```python
def main():
    fun_test()

if __name__ == '__main__':
    main()
```

结果展示

Task4

本任务的均在 week5 作业的基础上测试

每个装饰器展示添加的代码与结果

line_profiler

```
172    lp = line_profiler.LineProfiler()
173    @lp
```

```
main()
sys.stdout = open('print.log', mode = 'w',encoding = 'utf-8')
lp.print_stats()
```

print - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
Timer unit: 1e-07 s

Total time: 0.252963 s
File: e:\code\py_code\week8\week8_task4.py
Function: main at line 172

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
   172                                           @lp
   173                                           def main():
   174                                               """
   175                                               main函数
   176                                               """
   177         1      22631.0  22631.0      0.9      txt=fread_document(filename_0_test) #读取txt文件
   178         1         21.0     21.0      0.0      m=len(txt)
   179                                               #txt=fdelete_repetition_txt(txt) #删除重复项
   180         1          5.0      5.0      0.0      n=len(txt) #记录项数
   181                                               #print("处理前数据有%d项.\n处理重复数据后有%d项." %(m,n))
   182         1      13706.0  13706.0      0.5      data=fcut_txt(txt,n) #简单切割内容、地址、时间
   183         1         11.0     11.0      0.0      n=len(data)#更新项数
   184                                               #print(n)
   185         1       8790.0   8790.0      0.3      sentence = fdelete_url_data(data,n) #简单分词处理
   186         1     222327.0 222327.0      8.8      sentence = fclean_sentence(sentence,n) #正则表达式降噪
   187                                               #print(sentence)
   188         1        424.0    424.0      0.0      chars = ''.join(sentence)
   189                                               #print(chars)
   190                                               #T_c = Tokenizer(chars,'c')
   191                                               #print("文本总长度为%d"%T_c.len_all)
   192                                               #print('按字编码的编码规模为%d'%len(T_c.dic_chars))
   193                                               #T_w = Tokenizer(chars,'w')
   194                                               #print("文本总词数为%d"%T_w.len_all)
   195                                               #print('按词编码的编码规模为%d'%len(T_w.dic_chars))
   196         1      87933.0  87933.0      3.5      T_c = Tokenizer(chars,'c')
   197         1         14.0     14.0      0.0      lis_token=[]
   198         1          5.0      5.0      0.0      seq_len = 34
   199        11        278.0     25.3      0.0      for i in range(10):
   200        10        890.0     89.0      0.0          number = random.randint(0,len(sentence)-1)
   201        10       1122.0    112.2      0.0          lis_of_chars = T_c.tokenize(sentence[number])
   202        10       1332.0    133.2      0.1          tokens = T_c.encode(lis_of_chars)
   203        10        906.0     90.6      0.0          tokens = T_c.trim(tokens,seq_len)
   204        10         56.0      5.6      0.0          lis_token.append(tokens)
   205         1       3488.0   3488.0      0.1      print("10条文本Trim后的token为：")
   206        11        131.0     11.9      0.0      for i in lis_token:
   207        10      11913.0   1191.3      0.5          print(i)
   208         1        625.0    625.0      0.0      print('------------------------------')
   209         1        380.0    380.0      0.0      print("10条文本解码后为：")
   210        11        406.0     36.9      0.0      for i in lis_token:
```

memory_profiler

```
mp = memory_profiler.profile()
@mp
def fread_document(filename) -> list:
    """
```

```
Line #    Mem usage    Increment  Occurrences   Line Contents
=============================================================
   106     62.4 MiB     62.4 MiB           1   @mp
   107                                         def fread_document(filename) -> list:
   108                                             """
   109                                             按行读入txt文件并返回列表
   110                                             ->返回一维列表
   111                                             :filename:原始文档目录
   112                                             """
   113     62.4 MiB      0.0 MiB           1       print("----------正在导入数据----------")
   114     62.4 MiB      0.0 MiB           1       f=open(filename,encoding='UTF-8')
   115     62.4 MiB      0.0 MiB           1       line = f.readline().strip() #读取第一行,不用读入列表
   116     62.4 MiB      0.0 MiB           1       txt=[]
   117    693.9 MiB  -1045.9 MiB     1825758       while line:  # 直到读取完文件
   118    693.9 MiB   -434.0 MiB     1825757           line = f.readline().strip()  # 读取一行文件，包括换行符
   119    693.9 MiB  -1025.7 MiB     1825757           txt.append(line)
   120    693.9 MiB      0.0 MiB           1       f.close()  # 关闭文件
   121    693.9 MiB      0.0 MiB           1       if txt[-1]=='':
   122    693.9 MiB      0.0 MiB           1           txt.pop()
   123    693.9 MiB      0.0 MiB           1       print("----------数据导入完成----------")
   124    693.9 MiB      0.0 MiB           1       return txt
```

tqdm

```
if coding == 'c':
    self.len_all = len(chars)
    for char in tqdm.tqdm(chars):
        if char not in dic:
            dic[char] = code_number
            code_number += 1
```

进度条展示效果

```
PS E:\code\py_code> python -u "e:\code\py_code\week8\week8_task4.py"
----------正在导入数据----------
----------数据导入完成----------
 43%|                                                                                                  | 21555999/49964824 [00:03<00:05, 5376186.00it/s]
```

```
PS E:\code\py_code> python -u "e:\code\py_code\week8\week8_task4.py"
----------正在导入数据----------
----------数据导入完成----------
100%|                                                                                                  | 49964824/49964824 [00:09<00:00, 5236472.62it/s]
10条文本Trim后的token为：
[2494, 1289, 34, 1337, 440, 13, 2143, 419, 34, 82, 39, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[80, 1528, 85, 332, 333, 138, 1938, 259, 298, 63, 637, 691, 1787, 373, 367, 376, 372, 369, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Pysnooper

```
@pysnooper.snoop()
def main():
```

```
PS E:\code\py_code> `python -u "e:\code\py_code\week8\week8_task4.py"
Source path:... e:\code\py_code\week8\week8_task4.py
00:36:16.452027 call       181 def main():
00:36:16.454016 line       185     txt=fread_document(filename_0) #读取txt文件
----------正在导入数据----------
----------数据导入完成----------
New var:..... txt = ['[22.940469, 113.10337]\t若苦苦追求过却还是一无所获，又何必硬要去强求...702495408454620\tWed Apr 23 12:25:57 +0800 2014']
00:36:19.005723 line       186     m=len(txt)
New var:..... m = 1825756
00:36:21.372066 line       188     n=len(txt) #记录项数
New var:..... n = 1825756
00:36:23.594079 line       190     data=fcut_txt(txt,n) #简单切割内容、地址、时间
New var:..... data = [['[22.940469, 113.10337]', '若苦苦追求过却还是一无所获，又何必硬要...495408454620', 'Wed Apr 23 12:25:57 +0800 2014']]
00:36:28.746360 line       191     n=len(data)#更新项数
00:36:34.099854 line       193     sentence = fdelete_url_data(data,n) #简单分词处理
New var:..... sentence = ['若苦苦追求过却还是一无所获，又何必硬要去强求呢？不妨这样安慰自己：该是你的，躲也躲不过；不是...双待，才799元，刚刚成功预约了4月29日中午12点开放购买，
希望能顺利买到1部，分享预约地址']
00:36:40.214700 line       194     sentence = fclean_sentence(sentence,n) #正则表达式降噪
00:37:29.842005 line       196     chars = ''.join(sentence)
New var:..... chars = '若苦苦追求过却还是一无所获，又何必硬要去强求呢？不妨这样安慰自己：该是你的，躲也躲不过；不是你...卡双待，才799元，刚刚成功预约了4月29日中午12点开放购买，
希望能顺利买到1部，分享预约地址'
```

```
恭喜你中枪了，评论那po或赞的就必须在下面选一句话发po，出来混要玩
00:48:13.467472 line       218     for i in lis_token:
Modified var:.. i = [561, 85, 13, 873, 85, 13, 298, 85, 13, 1446, 85..., 102, 102, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
00:48:19.315995 line       219         T_c.decode(i)
睡了，梦了，乐了，醒了，睡了，梦了，没了～～～[PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD]
00:48:25.159409 line       218     for i in lis_token:
Modified var:.. i = [205, 69, 909, 117, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0..., 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
00:48:30.974528 line       219         T_c.decode(i)
好！巴士[PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD]
00:48:36.879447 line       218     for i in lis_token:
Modified var:.. i = [150, 227, 63, 521, 576, 769, 813, 255, 499, 0, ..., 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
00:48:42.738085 line       219         T_c.decode(i)
我在大桥四线终点站[PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD]
00:48:48.621277 line       218     for i in lis_token:
Modified var:.. i = [670, 647, 388, 2015, 437, 13, 205, 170, 171, 0,..., 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
00:48:54.484478 line       219         T_c.decode(i)
红色加蕾丝，好喜欢[PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD]
00:49:00.472927 line       218     for i in lis_token:
Modified var:.. i = [205, 301, 205, 301, 33, 414, 165, 204, 34, 242,...918, 85, 165, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
00:49:06.345715 line       219         T_c.decode(i)
好爱好爱你们 真的被这么有爱的电视剧融化了 [PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD]
00:49:12.414521 line       218     for i in lis_token:
Modified var:.. i = [403, 609, 527, 144, 1358, 13, 301, 61, 87, 1178..., 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
00:49:18.285657 line       219         T_c.decode(i)
生命诚可贵，爱情最易醉[PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD][PAD]
00:49:24.148785 line       218     for i in lis_token:
00:49:30.050328 line       220     """
00:49:35.870662 line       236     """
00:49:41.629848 return     236     """
Return value:.. None
Elapsed time: 00:13:31.373841
```