

文件空行，注释，行数统计

不知不觉你已经写了很多python代码了，代码全部都放在e盘下面的my_code文件夹中，突然突发奇想，想统计了一下总共写了多少行代码，包括空行和注释，要把代码的行数、空行数、注释行数都统计出来

1、找到e盘下的所有py文件

2、每个文件进行遍历，找总行数，空行数，注释行数都统计

空行就是\n

注释行 三开头，三点结尾 #号开头且都是首

1、每个统计数单写方法

```
  
import os  
  
#统计多行注释的注释行数  
  
def note_three(f):  
    t = False #开关关闭状态  
  
    count = 0 #总数初始为0  
  
    for i in f:  
  
        if i.strip().startswith('"""'): #当行是'''开头的，开始记行数  
  
            t = not t #开关打开  
  
            count += 1 #开头行和结尾行的计数  
  
            continue  
  
        if t == True:  
  
            count += 1 #注释中间的行计数  
  
    f.seek(0) #文件索引归初始位置  
  
    return count #返回多行注释的行数  
  
def note_one(f):  
    count=0 ##号注释统计初始值  
  
    for line in f:  
  
        if line.strip().startswith('#'): #统计所有以#号开头的行  
  
            count+=1 #每次加1  
  
    f.seek(0) #文件索引归初始位置  
  
    return count #返回单行注释的行数  
  
def null_num(f):
```

```

count=0 #空行统计初始值

for line in f :

    if not line.strip(): #如果空行，统计数加1

        count+=1

f.seek(0) #文件索引归初始位置

return count #返回空行行数


def row_num(f):

    count=0 #文件行数统计初始值

    for line in f :

        count+=1 #每变量一行，文件总行数加1

    f.seek(0) #文件索引归初始位置

    return count #返回文件总行数


path='.' #文件路径

files = os.listdir(path) #获取路径下文件list

count_dic = {} #结果存放处

for name in files: #变量文件

    if name.endswith('.py'): #文件以.py结尾，开始统计文件行数，空行，注释

        file = open(name, encoding='utf-8') #打开文件

        j_count=note_one(file)+note_three(file) #注释总数

        count=row_num(file) #行总数

        i_count=null_num(file) #空行总数

        count_dic[name]={'行数': count, '空行': i_count, '注释': j_count} #循环放入字典每个文件的各项信息

print(count_dic) #打印结果

```

2、减少遍历文件的次数

不知不觉你已经写了很多python代码了，代码全部都放在e盘下面的my_code文件夹中，突然突发奇想，想统计了一下总共写了多少行代码，包括空行和注释，要把代码的行数、空行数、注释行数都统计出来

1、找到e盘下的所有py文件

2、每个文件进行遍历，找总行数，空行数，注释行数都统计

空行就是\n

注释行 三点头，三点结尾 #号开头且都是首

```
import os
```

#统计多行注释的注释行数

```
def note_three(f):  
    t = False #开关关闭状态  
    count = 0 #总数初始为0  
    for i in f:  
  
        if i.strip().startswith('"""'): #当行是'''开头的，开始记行数  
            t = not t #开关打开  
            count += 1 #开头行和结尾行的计数  
            continue  
  
        if t == True:  
            count += 1 #注释中间的行计数  
    f.seek(0) #文件索引归初始位置  
  
    return count #返回多行注释的行数  
  
def count_file(f):  
    row_num=0 #总行数  
    note_one=0 #单行注释行数  
    null_num=0 #空行数  
    for index,line in enumerate(f): #遍历文件行内容，和行号  
        row_num += 1 #每遍历一行加1  
        if not line.strip(): #遇到空行加1  
            null_num += 1  
        elif line.strip().startswith('#'): #每遇到#号开头单行注释加1  
            note_one += 1  
    f.seek(0) #文件索引归0  
    return row_num,null_num,note_one #返回总行数，空行总数，单行注释总数  
  
path='.' #文件路径  
files = os.listdir(path) #获取路径下文件list  
count_dic = {} #结果存放处  
for name in files: #变量文件
```

```
if name.endswith('.py'): #文件以.py结尾, 开始统计文件行数, 空行, 注释

    file = open(name, encoding='utf-8') #打开文件

    row_num,null_num,note_one=count_file(file) #获取总行数, 空行总数, 单行注释总数

    note_num=note_one+note_three(file) #注释总数

    count_dic[name]={ '行数': row_num, '空行': null_num, '注释': note_num} #循环放入字典每个文件的各项信息

print(count_dic) #打印结果
```