Python大作业:文本格式转换和统计

一、任务简述

本任务利用正则表达式解析给定的《The Merchant of Venice》HTML网页文件，并将文件内容按Markdown格式存储至文件中，同时把统计结果写入文件中。

主要考察学生以下几个方面：

- 程序设计能力及Python编程模式的理解；
- 运算符、表达式、内置函数及序列结构的运用能力；
- Python分支结构、循环结构及函数设计的掌握情况；
- 运用正则表达式处理字符串的能力；
- Python读写文本文件。

二、背景：HTML和Markdown

2.1 HTML

超级文本标记语言（英文缩写：HTML）是为“网页创建和其它可在网页浏览器中看到的信息”设计的一种标记语言。HTML是一种标准，通过标记符号来标记要显示的网页中的各个部分。网页文件本身是一种文本文件，通过在文本文件中添加标记符，可以告诉浏览器如何显示其中的内容（如：文字、表格、图片如何显示等）。在浏览器中，单击右键，选择查看网页源代码，即可查看页面对应的HTML代码。

下面是一段HTML文本示例源代码

```
<p>
Act 1, Scene 1: <ahref="./merchant/merchant.1.1.html">Venice. A street.</a><br>
Act 1, Scene 2: <ahref="./merchant/merchant.1.2.html">Belmont. A room inPORTIA'S
house.</a><br>
Act 1, Scene 3: <ahref="./merchant/merchant.1.3.html">Venice. A publicplace.</a>
<br>
</p>
```

该内容在浏览器中显示如下

```
Act 1, Scene 1: Venice. A street.
Act 1, Scene 2: Belmont. A room in PORTIA'S house.
Act 1, Scene 3: Venice. A public place.
```

HTML标记也称HTML标签(HTML tag)，标签的详细含义可查阅HTML参考手册，此处只做简单介绍：

- HTML标记是由尖括号包围的关键词，比如<html>；
- HTML标记通常是成对出现的，比如 和 ；
- 标记对中的第一个标记是开始标记，第二个标记是结束标记；

- HTML标记英文字母不区分大小写，即<a>和<A>等同。

2.2 MARKDOWN

Markdown是一种轻量级标记语言，简单易学，用途广泛，易于与HTML进行转换。更加详细的可以参考 https://www.jianshu.com/p/1e402922ee32 "Markdown——入门指南"

三、程序说明

3.1 输入数据说明

附件包含document和data两个子目录：document目录存放输出(含示例)文件，data目录存放需处理的HTML文件。

data目录中的文件Merchant of Venice_ List of Scenes.html是一个**网页目录文件**，其网页中记录了各场剧集的相对路径；在data目录的子目录merchant中，包含所有的剧集文件，文件名形如**merchant.x.y.html**，其中的**x.y**代表**第x幕第y场**，共5幕20场。

3.2 程序的输入、输出及流程

程序首先询问网页目录文件Merchant of Venice_ List of Scenes.html所在的位置，然后根据输入的文件位置获取Merchant of Venice_ List of Scenes.html文件的内容，对于网页目录文件，主要提取所有剧集所在的位置：即提取所有形如**merchant.x.y.html**的字符串。

提取剧集所在的位置之后，对每一个剧集文件，分析其内容。分析分为两个部分：第1部分，从剧集文件中提取文本，并将其转换成Markdown的标记格式，最后存入一个文件中，具体转换请参考下面的**输出格式说明**。第2部分，统计所处理的所有网页文件中的各个HTML标签的出现次数，最后按标签出现的次数从多到少的顺序存入文件中。注意：对于成对的标签，如 仅记做 出现一次。

程序最终输出两个结果：一个是Markdown文件venice.MD，另一个是标签统计文件venice.Tag，这两个文件都是文本文件，保存在目录document下面。

3.3 输出格式说明

程序对网页目录文件Merchant of Venice_ List of Scenes.html和剧集文件merchant.x.y.html(共20个)进行处理，提取内容并存入名为venice.md的Markdown文件中：

1. 剧本名单独一行，且前后各空一行，被空行包围，作为一级标题，需在名称前加一个#，并以空格分隔。剧本名可以从目录文件Merchant of Venice_ List ofScenes.html中提取，通常在<title>标记中。
2. 幕号ACT单独一行，且前后各空一行，被空行包裹，作为二级标题，需在名称前加##，并以空格分隔，**仅在每一幕的开头添加**。可从剧集文件merchant.x.y.html中提取，如在merchant.1.1.html中，幕号ACT在头部位置，**Act 1, Scene 1**，需要抽取**Act 1**，生成 **## Act 1**。
3. 场名SCENE单独一行，且前后各空一行，被空行包裹，作为三级标题，需在名称前加###，并以空格分隔。场名可从剧集文件merchant.x.y.html中提取，通常在<h3>标记内。例如在merchant.1.1.html中<h3>SCENE I.Venice. A street.</H3>，需要提取出**SCENE I.Venice. A street.**，并生成 **### SCENE I.Venice. A street.**。
4. 人物名单独一行，使用两个星号(**)包裹，即**NAME**。人物名单在剧集文件merchant.x.y.html中，使用<a>标记，如在merchant.1.1.html中<A NAME=speech1>ANTONIO，此时NAME=speech1代表设置标记的NAME属性为speech1，需要把**ANTONIO**提取出来，并生成 ****ANTONIO****。

5. 台词根据提取文本分行，一个标记内的一段话即为一行，输出时在行尾添加两个空格(在换行符号之前)。台词在剧集文件merchant.x.y.html中，使用<a>标记。例如在merchant.1.1.html中，<A NAME=1>In sooth, I know not why I am so sad:</A>，此时NAME=1代表设置标记的NAME属性为1，代表台词序号。注：人物名与台词均使用<a>标记，区别在于NAME属性设置值不同，详细情况可将剧本幕的网页通过记事本的方式打开查看。需要把 **In sooth, I know not why I am so sad:** 抽取出来，直接生成 **In sooth, I know not why I am so sad: |** (|线前有两个空格，不包括最后的|线)。
6. 舞台说明又叫舞台提示，单独一行，且前后各空一行，使用1个星号(*)包裹，即*HINT*。舞台提示在剧集文件merchant.x.y.html中，使用标记。例如Enter ANTONIO, SALARINO, and SALANIO，需要把**Enter ANTONIO, SALARINO, and SALANIO**抽取出来，并生成 ***Enter ANTONIO, SALARINO, and SALANIO***。

3.4 输出文件示例

附件document目录下的文件Example for Venice.md展示了两场ACT剧本的输出格式，即附件data\merchant目录下的merchant.1.1.html和merchant.1.2.html解析转换后的结果。文件Example for Markdown to PDF.pdf是Example for The Merchant of Venice.md生成的pdf文件，这里不作为要求，仅作展示。文件Example for The Merchant of Venice.md的部分内容节录展示如下：

```
# The Merchant of Venice

## Act 1

### SCENE I. Venice. A street.

*Enter ANTONIO, SALARINO, and SALANIO*

**ANTONIO**

In sooth, I know not why I am so sad:
It wearies me; you say it wearies you;

...

*Exeunt*

### SCENE II. Belmont. A room in PORTIA'S house.

*Enter PORTIA and NERISSA*
```

附件document目录下的文件Example for Venice.tag记录了输出的部分结果。

四、程序功能函数

建议根据解析流程，可将程序划分为不同函数。在把函数连接成一个大程序之前，请仔细测试每个函数。

1. get_list_scene(file_path: str) -> list: 读取路径名称对应的HTML目录文件，并解析出各幕SCENE网页文件的路径,并以list类型作为函数返回。
2. get_scene_script(file_path: str) -> str: 读取路径名称对应的HTML文件，解析出该幕SCENE中的剧本，并将按格式存储的剧本以str格式作为函数返回。

3. write_script(file_name: str, content: str): 传入file_name，并以追加写入的方式打开，并将str写入。
4. get_dict_tags(file_path: str) -> dict: 读取路径名称对应的HTML文件，并解析出内含的所有标记名,注意不区分大小写,并以dict类型存储作为函数返回。
5. write_tagnum(file_path: str, tag_dict: dict): 传入file_name和分析得到的tag_dict，以写入的方式打开file_name，并将tag_dict的内容按照出现次数从多到少的顺序写入。

以上仅列出部分功能函数，供同学们参考，请同学们根据需要，自行修改或添加更多功能和函数。