导入：

import re

符号释义：

. 代表除了newline的字符(如果DOTALL指定，也包括newline)

^ 行起始

$ 行结束

\* 匹配前面字符0~N次(贪婪模式)

+ 匹配前面字符1~N次(贪婪模式)

? 匹配前面字符0或1次(贪婪模式)

\*? 匹配前面字符0~N次(懒惰模式)

+? 匹配前面字符1~N次(懒惰模式)

?? 匹配前面字符0或1次(懒惰模式)

{m} 精确匹配前面字符m次

{m,n} 匹配前面字符m~n次(贪婪模式)，可省略下限(默认0)和上限(默认无穷大)

{m,n}? 匹配前面字符m~n次(懒惰模式)

\ 跳脱字符，使特殊字符摆脱特殊意义或使字符成为特殊字符

[] 集合内字符选其一

[^] 除集合内的字符

tom|peter 二取其一

(…) 创建group，后续使用\NUMBER进行匹配引用

(?P<name>…)：创建命名式group，使用(?P=name)进行匹配引用

(?aiLmsux)：不创建group，代入一个或多个flags参数

(?:…)：匹配字符段(…)，但不创建分组，直接类似于…

(?aiLmsux-imsx:…)：对指定字符串(…)进行添加(?aiLmsux)或移除flags(-imsx)

(?#...)：注释内容

(?=…)：目标的后续必须匹配该字符段(…)，但不占位。常用于验证尾部

(?!...)：目标的后续必须不匹配该字符段(…)，不占位。常用于验证尾部

(?<=…)：目标的前置字符串必须匹配该字符段(…)，但不占位

(?<!...)：目标的前置字符串必须不匹配该字符段(…)，不占位

(?(id/name)yes-pattern|no-pattern)：当group number/name存在时，匹配yes-pattern，否则匹配no-pattern(类似于id/name?yes-pattern:no-pattern)

\NUMBER 分组匹配，从1开始，99结束

(?P=name)：同pattern下的命名式分组引用

\A 匹配字符串起始

\b 匹配单词起始或结尾的空字符串

\B 匹配非单词起始或结尾的空字符串

\d 匹配数字

\D 匹配非数字内容

\s 匹配空白字符([ \t\n\r\f\v])。其中，\t水平制表符、\v垂直制表符、\n换行符、\r回车符、\f换页符

\S 匹配非空白字符

\w 匹配单词字符([a-zA-Z0-9\_])

\W 匹配非单词字符

\Z 匹配字符串结尾

re模块常量(flags参数)：

A/ASCII：使用ASCII匹配代替默认的Unicode匹配

DEBUG：显示Debug信息

I/IGNORECASE：不关心大小写的匹配

L/Locale：适用于bytes模式，修改\b\w和IGNORECASE特征

M/MULTILINE：如未提供该flag，则全体字符串视为一行

S/DOTALL：使.匹配所有字符，包括newline

X/VERBOSE：使pattern以更为易读方式书写(忽略空白符和#后的字符)

re模块方法：

compile(pattern,flags=0)：编译一个正则匹配模式，返回Pattern对象

search(pattern,string,flags=0)：从源字符串中匹配第一个符合正则pattern格式的字符串，返回Match对象

match(pattern,string,flags=0)：从源字符串中起始位置匹配符合正则pattern格式的字符串(MULTILINE对之不起作用)，返回Match对象

fullmatch(pattern,string,flags=0)：整个源字符串完整匹配正则pattern

split(pattern, string, maxsplit=0, flags=0)：以正则pattern来分割源字符串，最大分割maxsplit次，其余以最后一个元素列出，返回的是list(若pattern为群组，则包含分割符部分)

findall(pattern,string,flags=0)：从源字符串中匹配所有符合正则pattern格式的字符串，返回匹配内容的list。其中，item为group(1,2…)的元组，当不含group时，item为group 0

finditer(pattern,string,flags=0)：从源字符串中匹配所有符合正则pattern格式的字符串，以iterator(惰性列表)形式返回

sub(pattern, repl, string, count=0, flags=0)：使用repl字符串或函数替换源字符串string中的pattern匹配部分，重复替换count次

subn(pattern, repl, string, count=0, flags=0)：使用repl字符串或函数替换源字符串中的pattern匹配部分，重复替换count次，并返回元组(result\_str,rep\_number)

escape(pattern)：使pattern中的特殊字符跳脱特殊含义

Pattern类常量：

groups：pattern中的组个数

pattern：pattern的匹配源内容

Pattern类方法：

search(string[,pos[,end-pos]])：使用正则从字符串任意处匹配，可指定字符串参与匹配的起始点和终点[pos,end-pos)

match(string[,pos[,end-pos]])：使用正则从字符串起始处开始匹配，可指定字符串参与匹配的起始点和终点[pos,end-pos)

fullmatch(string[,pos[,end-pos]])：使用正则匹配整个字符串，可指定字符串参与匹配的起始点和终点[pos,end-pos)

split(string[,pos[,end-pos]])：使用正则分割字符串，可指定字符串参与分割的起始点和终点[pos,end-pos)

findall(string[, pos[, endpos]])：从字符串中匹配所有符合正则的内容，可指定字符串参与匹配的起始点和终点[pos,end-pos)

finditer(string[, pos[, endpos]])：从字符串中匹配所有符合正则的内容，返回iterator，可指定字符串参与匹配的起始点和终点[pos,end-pos)

sub(repl,string,count=0)：使用repl替换字符串中的正则匹配部分，替换count次

subn(repl,string,count=0)：使用repl替换字符串中的正则匹配部分，替换count次，返回元组(new\_str,repl\_num)

Match类常量：

pos：原始字符串开始匹配位置，由Pattern方法参数pos指定

endpos：原始字符串的结束(开区间)匹配位置，由Pattern方法参数endpos指定

lastindex：匹配结束(最后一个”)”)组号的索引

lastgroup：最后一个匹配组的名称

re：匹配pattern的原文

string：用于匹配的原始字符串

Match类方法：

group([group1,group2,…])：返回对应组内容(0为完整pattern匹配内容)，当参数有多个时，返回包含多个组的元组

groups(default=None)：返回包含所有组的元组(1,2…)

groupdict(default=None)：返回key=组命名(?P<NAME>)，value=组的dictionary

start([group])：返回对应组的起始位置索引

end([group])：返回对应组的结束位置索引(开区间)

span([span])：返回对应组的元组(start,end)