import re

<https://www.runoob.com/python/python-reg-expressions.html#flags>

re中正则表达式都是以字符串的形式输入的，为了方便编译，往往在正则表达式的引号外加非转义原生字符符号r

**bool(re.match(pattern,string))** #检查字符串是否符合pattern

**matchObj=re.match(pattern,string,flags=0)**  #从字符串头开始匹配模式

matchObj.span(n) #输出匹配对象或捕获对象在整个字符串中的匹配位置

matchObj.start(n) #输出匹配对象或捕获对象在整个字符串中的起始位置

matchObk.end(n) #输出匹配对象或捕获对象在整个字符串中的结束位置

matchObj.group(n) #输出匹配对象或捕获对象，n为pattern中有效捕获符号（）数加1。group(0)为匹配到的整体字符串

matchObj.groups() #输出所有匹配对象或捕获对象

**searchObj=re.search(pattern, string, flags=0)** #扫描整个字符串并返回第一个成功的匹配

searchObj.span()

searchObj.group(n)

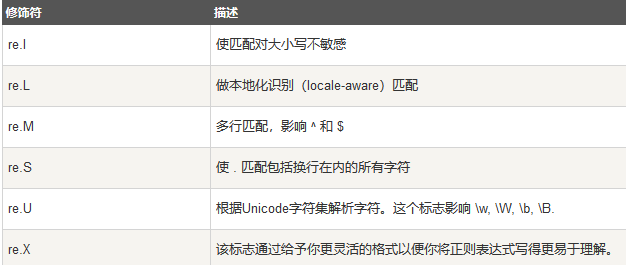
searchObj.groups()

re.match只匹配字符串的开始，如果字符串开始不符合正则表达式，则匹配失败，函数返回None；而re.search匹配整个字符串，直到找到一个匹配。

re.match和re.search都只匹配一次

**re.sub(pattern, repl, string, count=0, flags=0)** #用repl替换匹配到的字符串，repl可以是一个函数；count表示替换次数（缺省0表示全替换）

**flags修饰符**



欲同时使用多个修饰符，可通过或逻辑 | 来将它们并联

**pattern=re.compile(pattern,flags=0)**  #编译模式

**pattern.findall(string,start,end)** #根据编译的模式匹配所有子串，以数组形式返回

**re.split(pattern,string,maxsplit=0,flags=0)**

#利用匹配到的子串作为分隔符，切分原字符串并返回一个数组，maxsplit=0不限次数）

#如果找不到子串，则会返回原字符串而不进行分割

#如果分隔符处在字符串开头或结尾，会在切分结果中多出一个空字符””

#如果用括号在引号里把pattern括起来，切分结果中会连带分隔符一起输出

**pattern.findall(string, flags=0)** #找到字符串中所有符合pattern的字段

[**https://www.runoob.com/regexp/regexp-metachar.html**](https://www.runoob.com/regexp/regexp-metachar.html)

**特殊字符**

**^ 字符串开始符；用在括号中表示非逻辑**

**$ 字符串结尾符**

**\* 前面子表达式0次或多次**

**+ 前面子表达式1次或多次**

**? 如果前面是单个字符, 则前面子表达式0次或1次；如果前面是连续匹配的多个字符，则表示匹配到的个数尽可能少（如\*或+）**

**. 换行符外的任何单字符**

**\ 转义**

**| 或逻辑**

**[] 多字符或逻辑**

**{n} 匹配n次**

**{n,} 至少匹配n次**

**{n,m} 至少匹配n次，至多匹配m次**

**[\{}]**

**\t 制表符**

**\n 换行符**

**\r 回车符**

**\s 空格、制表符、换行符、回车符、换页符**

**\S \s的非逻辑**

**\f 换页符**

**'[!"#$%&\'()\*+,-./:;<=>?@[\\]^\_`{|}~“”]' 所有标点符号**

**\b 匹配单词边界**

**\B 匹配非单词边界**

**\d [0-9]**

**\D [^0-9]**

**[A-Za-z] 所有字母**

**\w [A-Za-z0-9\_]**

**\W [^A-Za-z0-9\_]**

**（）等价于<>**

**(pattern) #括号运算，捕获括号中的pattern；用圆括号会使相关的匹配会被缓存，此时可用?:放在第一个选项前来消除这种副作用。**

**(pattern)\num #重复连续获取括号中的pattern共num次，num为1~99的整数**

**(?:pattern) #捕获括号中的pattern但是不保存结果（用re.search搜索不到）**

**xxx (?=pattern) #正向肯定预查，捕获以pattern结尾的内容xxx，pattern的匹配不会被保存**

**xxx (?!pattern) #正向否定预查**

**(?<=pattern) xxx #反向肯定预查，捕获以pattern开头的内容xxx，pattern的匹配不会被保存**

**(?<!pattern)xxx #反向否定预查**



**运算优先级**



**模式修正符**

**i**:不区分大小写

**g**:全局匹配

**m**:无视换行符进行匹配

**s**:将字符串视为单行

**x**:忽略空白符号（空格、制表符等）

**A**:强制从目标字符串开头匹配

**Z**:强制从目标字符串尾匹配

**D**:如果使用$限制结尾字符,则不允许结尾有换行

**U**:只匹配最近的一个字符串;不重复匹配



[**https://www.cnblogs.com/yuanwanli/p/5838918.html**](https://www.cnblogs.com/yuanwanli/p/5838918.html)

[**https://blog.csdn.net/weixin\_30786617/article/details/96441169**](https://blog.csdn.net/weixin_30786617/article/details/96441169)