在python中需要使用正则表达式时，通常需要一个模块，这个模块就是re，下来简单的介绍下re的一些方法。

## match()方法

re.match 是用来进行增则匹配检查的方法，若字符串匹配正则表达式，则match方法返回匹配对象，负责返回None，而不是空字符串。re.match（）能够匹配以xxxx开头的字符串。

匹配对象具有group方法，用来返回字符串的匹配部分。

span方法用来返回匹配的范围。

import re

content = 'Hello, my name is Frank'

result = re.match('^He\w\w\w', content)

print(result)

print(result.group())

print(result.span())

结果如下：

<re.Match object; span=(0, 5), match='Hello'>

Hello

(0, 5)

## search()方法

match()方法是从字符串的开头开始匹配，一旦开头不匹配，那么整个匹配就失败了。

search()方法在匹配时会扫描整个字符串，返回第一个匹配成功的内容，如果扫描完后还没找到，则返回None。

import re

content = 'Hello, my name is Frank'

result = re.match('F\w\w\w', content)

print(result)

result2 = re.search('F\w\w\w', content)

print(result2)

result3 = re.search('U\w\w\w', content)

print(result3)

结果：

None

<re.Match object; span=(18, 22), match='Fran'>

None

## findall()方法

如果想要获取正则表达式匹配的所有内容，显然上述两种方法不适合，这时就需借助findall（）方法了，该方法会搜索整个字符串，并返回所有符合匹配规则的内容。

1

import re

content = 'Hello, my name is Frank'

result1 = re.match('\w+', content)

print(result1)

result2 = re.search('\w+', content)

print(result2)

result3 = re.findall('\w+', content)

print(result3)

结果：

<re.Match object; span=(0, 5), match='Hello'>

<re.Match object; span=(0, 5), match='Hello'>

['Hello', 'my', 'name', 'is', 'Frank']

## sub()方法

除了使用正则表达式提取信息外，还可以借助其用来修改文本。

sub()方法就可将匹配到的数据进行替换。

1

2

import re

content = 'Hello my name is Frank'

result = re.sub('\s','\_', content) #用\_来替换空格，\s代表空格，tab文件

print(result)

结果：

Hello\_my\_name\_is\_Frank

## compile()方法

compile()方法可以将正则表达式编译成正则表达式对象，以便在后面的匹配中复用。

例如，将三个日期中的时间去掉，借助sub方法，由于正则表达式相同，没必要写三个，可以用compile方法将正则表达式编译成一个正则表达式对象，进行复用

import re

content1 = '2019-02-28 20:00'

content2 = '2019-03-01 20:00'

content3 = '2019-03-02 20:00'

pattern = re.compile('\d{2}:\d{2}')

result1 = re.sub(pattern, '', content1)

result2 = re.sub(pattern, '', content2)

result3 = re.sub(pattern, '', content3)

print(result1)

print(result2)

print(result3)

结果：

2019-02-28

2019-03-01

2019-03-02