链滴

Python 正则表达式常用方法示例汇总

作者：hljhrbeu
原文链接：https://ld246.com/article/1545707212418
来源网站：链滴

RE模块的方法

RE 正则表达式 写法（词组整体出现次数的写法）

相比常规的正则表达式，词组的写法是添加?:这个特殊标志

最外围（）是需要截取的字符串

此问题参考网站

```
>>>pattern_index=re.compile(r'((?:qsyn\-)?logstash[\-a-z0-9\.]+)') ### 用来匹配带有qsyn- 和带qsyn- 的字符串
>>>pattern_index.findall("green open qsyn-logstash-logs-2018.12.20 zdmqszKURqGmnfxr6CkwQ ")
['qsyn-logstash-logs-2018.12.20']
>>>pattern_index.findall("green open logstash-logs-2018.12.20 zdmqszKURqGmnfUxr6CwQ ")
['logstash-logs-2018.12.20']
>>>pattern_index=re.compile(r'((?:qsyn\-)?logstash[\-a-z0-9\.]+)0')
>>>pattern_index.findall("green open logstash-logs-2018.12.20 zdmqszKURqGmnfUxr6CwQ ")
['logstash-logs-2018.12.2'] #### 注意表达式的区别，并且外围（） 的区别

re.compile(pattern) ## 将正则表达式pattern 编译成 pattern对象
parttern=re.compile(r'[0-9]+\.com$')

## match (默认从头部开始匹配，除非特殊指定起始位置)返回 字符串对象
a=parttern.match('123.com') ## match 匹配以后返回 字符串对象
print a
<_sre.SRE_Match object at 0x10cd838b8>
a.group()
'123.com'
```

原文链接：Python 正则表达式常用方法示例汇总

```
## 不区分大小写 和 分组示例
pattern=re.compile(r"([a-z]+) ([a-z]+)",re.I) ### 不区分大小写；分组
m=pattern.match("hello wWWWd")
m.group()
'hello wWWWd'
m.group(1)
'hello'
m.group(2)
'wWWWd'
m.span(0)    # 返回匹配成功的整个子串的索引
(0, 11)


## search 较match比较，可以匹配任意位置，匹配一次成功后则完成;注意以下区别
m=pattern.match("123hello worLD")
print m
None
m=pattern.search("123hello worLD")
print m
<_sre.SRE_Match object at 0x10d038250>
```

findall 方法： 以列表形式查找所有匹配的结果，如果有则返回列表

finditer 方法：类似于findall，不过返回的是iter迭代对象

```
m=pattern.findall("123hello worLD 23423ECH WORLDBEST")
print m
[('hello', 'worLD'), ('ECH', 'WORLDBEST')
pattern2=re.compile(r"[a-z]+",re.I)
n=pattern2.findall("123hello worLD 23423ECH WORLDBES")
print n
['hello', 'worLD', 'ECH', 'WORLDBES']
### find iter
t=pattern2.finditer("123hello worLD 23423ECH WORLDBES")
for i in t:
    print i.group()

hello
worLD
ECH
WORLDBES

t=pattern2.finditer("123hello worLD 23423ECH WORLDBES")
for i in t:
    print i

<_sre.SRE_Match object at 0x10cd83988>
<_sre.SRE_Match object at 0x10cd839f0>
<_sre.SRE_Match object at 0x10cd83988>
<_sre.SRE_Match object at 0x10cd839f0>
```

split 方法： 可以理解成取反，即将匹配到的字符串的分隔符（分割符串）以列表的形式返回.常用于多个分隔符分割字符串时使用。

```
p=pattern2.split("123hello worLD 23423ECH WORLDBES")
print p
['123', ' ', ' 23423', ' ', '']
n=pattern2.findall("123hello worLD 23423ECH WORLDBES")
print n
['hello', 'worLD', 'ECH', 'WORLDBES'
>>>print test
a-b-c.d.e.f.g
>>>print re.split(r"[.\-]",test)
['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

sub 方法： 用于替换，shell中的sed

sub(repl, string[, count]) :

```
pattern3=re.compile(r"[0-9]+")
s="hello123world456"
p=pattern3.sub("AAAAAA",s) ## 用AAAAAA 替换所有匹配到的
print p
helloAAAAAAworldAAAAAA
def fun(m):
    return "hi"+m.group()
p=pattern3.sub(fun,s)  ## 利用def 的 fun函数将匹配到的结果进行处理后替换。注意group() 的使

print p
hellohi123worldhi456
p=pattern3.sub(fun,s,1)  ## 只匹配一次
print p
hellohi123world456
```

subn方法：与sub的方法行为类似，也用于替换:返回一个元组（字符串，匹配数）

```
p=pattern3.subn("AAAAAA",s)
print p
('helloAAAAAAworldAAAAAA', 2)
p=pattern3.subn(fun,s)
print p
('hellohi123worldhi456', 2)
p=pattern3.subn(fun,s,1)
print p
('hellohi123world456', 1)
```