2.6给你想要的正则表达式

2018年4月12日 10:03

1. KV编程论：各种模式匹配都只是代数
2. 匹配一次还是多次
3. 如果想分组匹配怎么办
4. 匹配要不要贪心一点
5. 正则匹配可以直接换掉内容吗？
6. 案例：找找她的联系方式
7. 案例：登录验证正则版
8. 项目：51备忘录v0.27

2.6.1. KV编程论：各种模式匹配都只是代数

2018年4月12日　　10:04

KV编程论：

正则 -> WHY, WHAT, HOW

WHY:

从大量文本中查找规则字符串，比字符串各种查找都快，利用c语言的匹配引擎，广泛用于各种搜索，查找，爬虫

what:

正则 -> 代数，变量替换（用一些规定好的符号去匹配有规则的文本）

在线工具：

https://www.regexpal.com/

http://tool.oschina.net/regex

语法说明:

表 1. 正则表达式语法

符号	意义	例子
.	表示任意字符，如果说指定了 DOTALL 的标识，就表示包括新行在内的所有字符。	
^	表示字符串开头。	
$	表示字符串结尾。	‘ test ’ 可以匹配 ‘ test ’ 和 ‘ testtool ’ ，但 ‘ test$ ’ 只能匹配 ‘ test ’ 。
*, +, ?	添加到要匹配的目标后面， ’*’表示匹配 0 个或多个， ’+’表示匹配 1 个或多个， ’?’表示匹配 0 个或1个	‘ abc* ’ 可以匹配 ‘ abc ’ 或者 ‘ abcc ’ 或者 ‘ abcccc ’ 等等。 但 ‘ abc? ’ 只匹配 ‘ abc ’ ‘ abcc

		’ ‘ abcccc ’ 中的 abc。
*?, +?, ??	在上面的结果中只取第一个	<*> 会匹配’<H1>title</H1>’整个字符串（贪婪匹配），使用 *? 可以只找出 <H1>（非贪婪匹配）
{m}	对于前一个字符重复 m 次	a{6} 匹配 6 个’a’
{m,n}	对于前一个字符重复 m 到 n 次	a{2,4} 匹配 2-4 个 a，a{2,} 匹配 2 个以上 a，a{,4} 匹配 4 个以下 a
{m,n}?	对于前一个字符重复 m 到 n 次，并且取尽可能少的情况	在字符串’aaaaaa’中，a{2,4} 会匹配 4 个 a，但 a{2,4}? 只匹配 2 个 a
\	对特殊字符进行转义，或者是指定特殊序列	
[]	表示一个字符集	[abc] 会匹配字符 a，b 或者 c，[a-z] 匹配所有小写字母，[a-zA-Z0-9] 匹配所有字母和数字，[^6] 表示除了 6 以外的任意字符
|	或者，只匹配其中一个表达式	A|B，如果 A 匹配了，则不再查找 B，反之亦然
(…)	匹配括号中的任意正则表达式	
(?#...)	注释，忽略括号内的内容	
(?= …)	表达式’…’之前的字符串	在字符串’pythonretest ’ 中 (?=test) 会匹配’pythonre ’
(?!...)	后面不跟表达式’…’的字符串	如果’ pythonre ’ 后面不是字符串’ test ’，那么 (?!test) 会匹配’ pythonre ’
(?<= …)	跟在表达式’…’后面的字符串符合括号之后的正则表达式	正则表达式’(?<=abc)def ’ 会在’abcdef ’ 中匹配’def ’
(?<!...)	括号之后的正则表达式不跟在’…’的后面	

(?aiLmsux)	(使用一个或多个字符 'a', 'i', 'L', 'm', 's', 'u', 'x'.)分别对应这些正则标志符号: re.A (ASCII-only matching), re.I (ignore case), re.L (locale dependent),re.M (multi-line), re.S (dot matches all), re.U (Unicode matching), and re.X (verbose)	
(?:...)	正则括号的非捕获版本. 匹配括号内的任何正则表达式，但是组里匹配的子字符串在执行了匹配或者被之后的规则引用后将不能被获取	
(?imsx-imsx:...)	(使用零个或多个字符 'i', 'm', 's', 'x', '-'可选) 表示设置或去除相应的标志符号: re.I (ignore case), re.M(multi-line), re.S (dot matches all), and re.X (verbose), for the part of the expression. (The flags are described in Module Contents.) *New in version 3.6.*	
(?P<name>...)	和用括号的正则类似，但是name可以关联匹配到的字符串，作为分组名称	(?P<quote>['"]).*?(?P=quote) (匹配单引号或双引号的字符串):
(?P=name)	与上面命名的用法连用，用来代指匹配到的字符串	
(?(id/name)yes-pattern|no-pattern)	如果给定的组id或name存在，就通过yes-pattern去匹配，否则通过no-pattern匹配 no-pattern是可选的，可以去掉.	(<)?(\w+@\w+(?:\.\w+)+)(?(1)>|$) 匹配 '<user@host.com>' 和 'user@host.com', 但不能匹配 '<user@host.com' 或'user@host.com>'.

来自 <https://docs.python.org/3/library/re.html>

包含’ \ ’ **(转义符号)** 的特殊序列的意义如表 2-2：

表 2. 正则表达式特殊序列

特殊表达式序列	意义
\A	只在字符串开头进行匹配。
\b	匹配位于开头或者结尾的空字符串
\B	匹配不位于开头或者结尾的空字符串
\d	匹配任意十进制数，相当于 [0-9]
\D	匹配任意非数字字符，相当于 [^0-9]
\s	匹配任意空白字符，相当于 [\t\n\r\f\v]
\S	匹配任意非空白字符，相当于 [^ \t\n\r\f\v]
\w	匹配任意数字和字母，相当于 [a-zA-Z0-9_]
\W	匹配任意非数字和字母的字符，相当于 [^a-zA-Z0-9_]
\Z	只在字符串结尾进行匹配

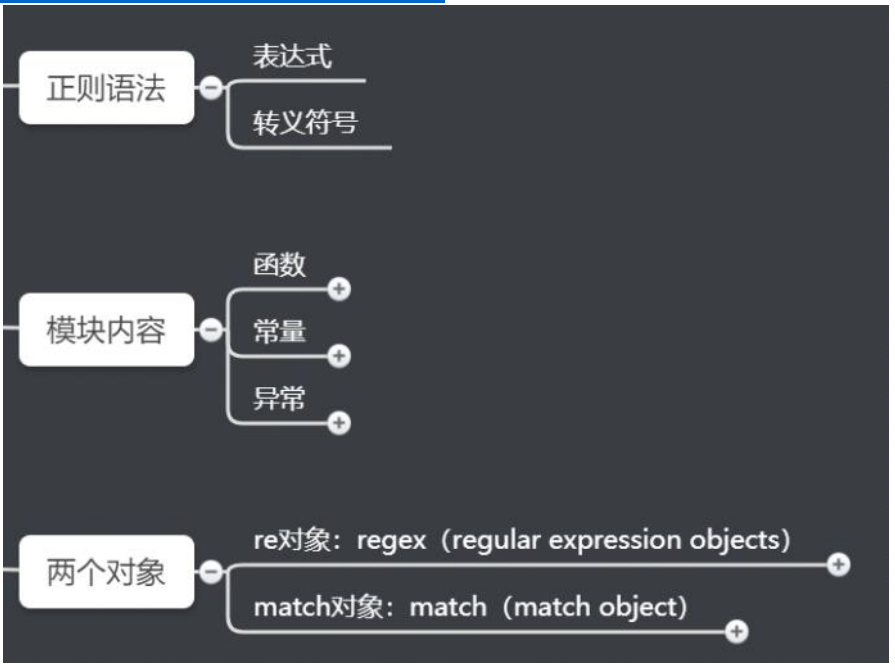
来自 <https://www.ibm.com/developerworks/cn/opensource/os-cn-pythonre/index.html>

2.6. 2. 匹配一次还是多次

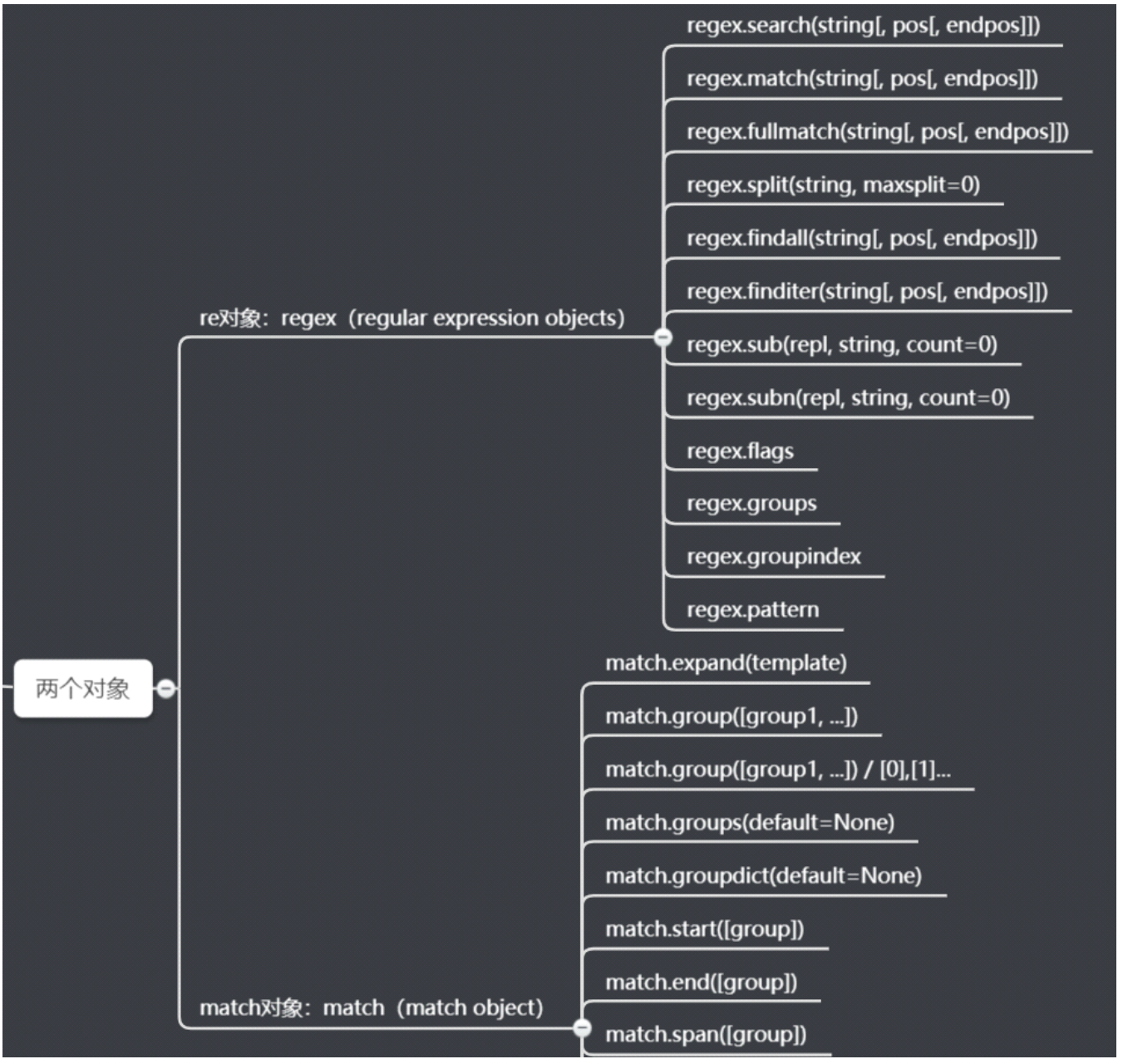
2018年4月12日 10:04

re模块使用说明:（hit and run）

https://docs.python.org/3/library/re.html




模块内容
函数
re.compile(pattern, flags=0) --regex
re.search(pattern, string, flags=0)¶ --match
re.match(pattern, string, flags=0) --match
re.fullmatch(pattern, string, flags=0) --match
re.split(pattern, string, maxsplit=0, flags=0) -- list
re.findall(pattern, string, flags=0) --list
re.finditer(pattern, string, flags=0) --iterator
re.sub(pattern, repl, string, count=0, flags=0) --str
re.subn(pattern, repl, string, count=0, flags=0)¶ --tuple
re.escape(pattern) --characters
re.purge()
常量
re.A/re.ASCII 让\w, \W, \b, \B, \d, \D, \s 和 \S仅执行ASCII-匹配
re.DEBUG 展示debug信息
re.I/IGNORECACE 忽略大小写
re.L/LOCALE 表示特殊字符集 \w, \W, \b, \B, \s, \S 依赖于当前环境
re.M/MULTILINE 多行模式
re.S/DOTALL 即为‘ .’并且包括换行符在内的任意字符（‘ .’不包括换行符)
re.X/VERBOSE 为了增加可读性，忽略空格和‘ #’后面的注释
异常



1- import re

2- re.compile()创建一个正则对象regex，一个变量多次使用

```
regex = re.compile(pattern)  # 使用regex对象，推荐，应用更灵活
result = regex.match(string)
==
match = re.match(pattern, string)  # 使用match对象
```

3- 使用regex查找一个字符串，返回被匹配的对象

4- 调用匹配对象的group方法，返回实际匹配的文本

*, +, ?	'*'表示后面可跟 0 个或多个字符，'+'表示后面可跟 1 个或多个字符，'?'表示后面可跟 0 个或1字符	‘ abc* ’ 可以匹配 ‘ abc ’ 或者 ‘ abcc’ 或者 ‘ abccccc’ 等等。
{m}	对于前一个字符重复 m 次	a{6} 匹配 6 个'a'
{m,n}	对于前一个字符重复 m 到 n 次	a{2,4} 匹配 2-4 个 a，a{2,} 匹配 2 个以上 a，a{,4} 匹配 4 个以下 a
[]	表示一个字符集	[abc] 会匹配字符 a，b 或者 c，[a-z] 匹配所有小写字母，[a-zA-Z0-9] 匹配所有字母和数字，[^6] 表示除了 6 以外的任意字符
|	或者，只匹配其中一个表达式	A|B，如果 A 匹配了，则不再查找 B，反之亦然

re.**compile**(*pattern, flags=0*)

Compile a regular expression pattern into a regular expression object

把正则表达式的模式和标识转化成正则表达式对象，供 match() 和 search() 这两个函数使用。

re 所定义的 flag 包括：

re.I 忽略大小写

re.L 表示特殊字符集 \w, \W, \b, \B, \s, \S 依赖于当前环境

re.M 多行模式

re.S 即为’ . ’并且包括换行符在内的任意字符（’ . ’不包括换行符）

re.X 为了增加可读性，忽略空格和’ # ’后面的注释

```
a = re.compile(r"""\d +  # the integral part
                   \.    # the decimal point
                   \d *  # some fractional digits""", re.X)
b = re.compile(r"\d+\.\d*")
```

re.**search**(*pattern, string, flags=0*)

Scan through *string* looking for the first location where the regular expression *pattern* produces a match, and return a corresponding match object.

```
>>> pattern = re.compile("a")
>>> pattern.search("abcde")      # Match at index 0
>>> pattern.search("abcde", 1)   # No match
```

re.**match**(*pattern, string, flags=0*)

If zero or more characters at the beginning of *string* match the regular

expression *pattern*, return a corresponding match object.
Note that even in MULTILINE mode, re.match() will only match at the beginning of the string and not at the beginning of each line.
If you want to locate a **match anywhere in *string***, use search() instead (see also search() vs. match()).

来自 <https://docs.python.org/3/library/re.html>

2.6.3. 如果想分组匹配怎么办

2018年4月12日 10:04

()	括号分组
|	管道符号匹配多个分组
?	选择出现0次或1次
re.x	换行，注释

findall

- 有分组，返回元组列表
- 无分组，返回字符串列表

split

- 返回用正则分割后的列表

案例：

ip地址

简单但错误版本：(\d{1,3}\.){3}\d{1,3}

正确的IP地址：((2[0-4]\d|25[0-5]|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5]|[01]?\d\d?)

2.6.4. 匹配要不要贪心一点

2018年4月12日 10:04

{m,n}?	对于前一个字符重复 m 到 n 次，并且取尽可能少的情况	在字符串'aaaaaa'中，a{2,4} 会匹配 4 个 a，但 a{2,4}? 只匹配 2 个 a
.* 与？的搭配	默认贪心，有多少要多少，？给加上限制，表示非贪心匹配， ？把*匹配的字符限制到最少	r'''(.*)''' 与r'''(.*?)'''

- *? 重复任意次，但尽可能少重复
- +? 重复1次或更多次，但尽可能少重复
- ?? 重复0次或1次，但尽可能少重复
- {n,}? 重复n次以上，但尽可能少重复

来自 <http://www.cnblogs.com/graphics/archive/2010/06/02/1749707.html>

2.6.5. 正则匹配可以直接换掉内容吗？

2018年4月12日 10:05

new_long_str = re_obj.sub(new_str, old_long_str)

- 添加分组\1\2\3...
- 替换字符串中间空格
- 格式转换

2.6.6. 案例：找找她的联系方式

2018年4月12日 10:05

手机号,电话号:

手机号码傻瓜版: ^1\d{10}$

电话号码必备区号版：\d{3}-\d{8}|\d{4}-\d{7}

匹配形式如 0511-4405222 或 021-87888822

邮箱:

电子邮件的验证：/(\w+@(\w+\.)+\w{2,3})?/

验证Email地址：^\w+[-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$

验证Email地址：/.+@.+\.[a-z]+/

身份证:

身份证号: ^(\d{15}|\d{17}(\d|X))$

2.6.7. 案例：登录验证正则版

2018年4月12日 10:05

用户名或密码的一些潜规则

- 字符大小写
- 长度限制
- 数字范围
- 手机
- 邮箱

常用正则表达式

- 匹配腾讯QQ号：[1-9][0-9]{4,}腾讯QQ号从10000开始
- 只能输入汉字：^[\u4e00-\u9fa5]{1,8}$
- 只能输入由数字和26个英文字母组成的字符串：“^[A-Za-z0-9]+$”
- 验证用户密码:“^[a-zA-Z]\w{7,17}$”正确格式为：以字母开头，长度在8-18之间， 只能包含字符、数字和下划线。

2.6.8. 项目：51备忘录v0.27

2018年4月12日 10:05

boss: 上次你给的备忘录程序，随着记录内容越来越多，我发现不够用了，比如我想快速统计一些数据，比如我想批量改改原来写错的内容，咋整?

小8：听说过正则表达式不?

boss:??

添加正则功能

- 从信息中提取关键词
  - findall
- 修改内容:
  - sub
- 验证输入
  - match

扩展功能:

- 场景1：根据已有数据：统计本月共多少人面试，整理手机号列表
  - 4.1日，共有4人面试，手机号分别是

13812345678，15112345678，13812345678，15112345678

- 4.5日，共有6人面试
  13812345678，15112345678，13812345678，15112345678，13812345678，15112345678
- 4.7日，共有3人面试13812345678，15112345678，13812345678
- 4.8日，共有5人面试
  15112345678，13812345678，15112345678，13812345678，15112345678
- 4.30日，共有6人面试
  13812345678，15112345678，13812345678，15112345678，13812345678，15112345678

- 场景2：有如下的多条备忘记录，请完成后续功能开发

  memo_text = '''
  1.1 去找小8写个程序
  1.2 记一下王总的电话 13912345678
  1.3 修改Python程序的bug
  1.4 路上买二斤西红柿，遇见卖鸡蛋的就买一斤
  1.5 事情太多，忘了今天要干啥
  '''

  修改里面的日期格式为几月几日，比如1.1 改为 1月1日

- 场景3：
  - 添加登陆验证功能，只限于个人使用