Python正则表达式

张敏

18/1/2

正则表达式

基础概念

正则表达式是一种可以用于模式匹配和替换的工具，可以让用户通过使用一系列的特殊字符构建匹配模式，然后把匹配模式与待比较字符串或文件进行比较，根据比较对象中是否包含匹配模式，执行相应的程序。

正则表达式

寻找字符串中的姓名和电话号码

rawdata = “555-1239Moe Szyslak(636) 555-0113Burns, C.Montgomery555-6542Rev. Timothy Lovejoy555 8904Ned Flanders636-555-3226Simpson,Homer5553642Dr. Julius Hibbert ”

正则表达式

Python正则表达式处理函数

1. match：re.match(pattern, string)

从首字母开始开始匹配，string如果包含pattern子串，则匹配成功，返回Match对象，失败则返回None

2. search：re.search(pattern, string)

若string中包含pattern子串，则返回Match对象（第一个），否则返回None

3. findall：re.findall(pattern, string)

返回string中所有与pattern相匹配的全部字串，返回形式为数组。

4. sub：re.sub(old, new, string)　替换操作

正则表达式

严格的字符匹配示例

- import re
- example_obj = "1. A small sentence. - 2. Another tiny sentence.”
- re.findall('sentence',example_obj)
- re.search('sentence',example_obj)
- re.sub('sentence','SENTENCE',example_obj)
- re.match('.*sentence',example_obj)

正则表达式

正则表达式的广义化

正则表达式的威力来源于能够编写灵活及广义化的查询条件

- re.findall('small',example_obj)
- re.findall('s.all',example_obj)
- re.findall('s[a-z]all',example_obj)
- re.findall('small|tiny',example_obj)

常用广义化符号

- 英文句号“.”：能代表除换行符“\n”任意一个字符；
- 字符类“[]”：被包含在中括号内部，任何中括号内的字符都会被匹配；
- 管道“|”：该字符被视为OR操作；

正则表达式

正则表达式的广义化字符

部分有特殊含义的符号

\w	数字和字母字符：[0-9a-zA-Z]
\W	与\w反义
\s	空白字符
\S	非空白字符
\d	数字：[0-9]
\D	非数字：[^0-9]
\b	单词的边界
\B	非单词边界

正则表达式

正则表达式的广义化字符

- example_obj = '1. A small sentence. -2. Another tiny sentence.’
- re.sub('\\d','kkk','abc12de')
- re.sub('[0-9]','kk','abc12de')
- re.sub('\w','kk','abc,12de')
- re.sub('\w{2}','kk','abcbe 12de')
- re.findall('[b-z]+',example_obj)
- re.findall('\\b[b-z]+\\b',example_obj)

正则表达式

正则表达式的广义化字符

Python正则表达式里的量化符

?	前面的元素是可选的，并且最多匹配1次
*	前面的元素会被匹配0次或多次
+	前面的元素会被匹配1次或多次
{n}	前面的元素会正好被匹配n次
{n,}	前面的元素至少会被匹配n次
{n,m}	前面的元素至少匹配n次，至多匹配m次

正则表达式

Python正则表达式里的量化符

- import re
- example_obj = "1. A small sentence. - 2. Another tiny sentence."
- re.findall("A.+sentence",example_obj)
- re.findall("A.?sentence",example_obj)
- re.findall("A.+?sentence",example_obj)
- re.findall("A.*sentence",example_obj)
- re.findall("A?sentence",example_obj)
- re.findall("A*sentence",example_obj)

正则表达式

练习1：寻找字符串中的姓名和电话号码

rawdata = “555-1239Moe Szyslak(636) 555-0113Burns, C.Montgomery555-6542Rev. Timothy Lovejoy555 8904Ned Flanders636-555-3226Simpson,Homer5553642Dr. Julius Hibbert”

- import re
- re.findall("[a-zA-Z.,]{2,}",rawdata)
- re.findall("[\d()\-]{0,6}\d{3}[-]?\d{4}",rawdata)

正则表达式

练习2：找出以下信息中的电话号码

J. Doe: 248-555-1234

B. Smith: (313) 555-1234

A. Lee: (810)555-1234

M. Jones: 734.555.9999

正则表达式

练习3：用一个正则表达式匹配出以下网址

http://www.forta.com/blog

https://www.forta.com:80/blog/index.cfm

http://www.forta.com

http://ben:password@www.forta.com

http://localhost/index.php?ab=1&c=2

http://localhost:8500/

Thank you!

泰迪科技：www.tipdm.com
热线电话：40068-40020