python中使用正则表达式

2015年10月3日 星期六　　上午10:07

一、简介

在python中有个库re，可以使用正则表达式，做匹配等。

二、使用方法

1、名词定义

原字符串：str

正则表达式：pattern

匹配上的字符串：sub_str

2、match和search功能

做pattern与str的匹配，用re库中的match或search函数

A 用法

```
regrex = re.match(pattern, str)
regrex = re.search(pattern, str)
```

例如：

```
regrex = re.match('dog', 'dog cat dog')
```

返回值regrex里面包含了匹配的信息，详见3

B match与search区别

Match 是匹配原字符串str的开头，也就是要求原字符串str从第一个字符开始，就能匹配上。

相当于str与sub_str的首字符是一个。

如"dog"匹配 "dog cat pig"

Search 是搜索原字符串str内部，只要原字符串str中sub_str就能匹配上。

相当于不要求str与sub_str的首字符是一个。

如"cat" 匹配 "dog cat pig"

3、匹配的返回值处理

使用match、search函数匹配字符串之后，如果匹配失败则match、search函数的返回值为None；如果匹配成功，则返回一个对象，这个对象包含了一些关于sub_str的信息，用下面函数可以打

出来。

A span()

span()

sub_str在str中的下标取值范围

如"dog"匹配 "dog cat pig" span是(0，3)，"cat"匹配 "dog cat pig" span是(4,7)

span(n)

如果pattern中有多个元组。如，'(.*) are (.*?) than (.*)' 包含三个元组：(.*)、(.*?)和(.*)

span(n) 可以返回第n个元组匹配上的sub_str 的下标

返回的是第n个元组的下标范围

span(n) n=0的时候，和span()是一样的，返回全部的sub_st的下标范围,（也就是说span()函数的默认值是n=0）

B group()

group()

返回匹配上的sub_str

group(n)

原理和span一样，返回第n个元组的sub_str

group(n) n=0的时候，和group()是一样的

也可以支持多个n的形式，如group(n,m)，是把group(n)与group(m)的结果放在一起

C start、end

span函数会返回下标范围，start、end返回的是下标的具体值，span()=(start(), end())

像span一样，也支持start(n)、end(n)

其他功能详见，参考资料2，有更高级详细的介绍

4、compile函数

compile可以预先编译，代替re

```
4  # 将正则表达式编译成Pattern对象
5  pattern = re.compile(r'hello')
6
7  # 使用Pattern匹配文本，获得匹配结果，无法匹配时将返回None
```

```
8  match = pattern.match('hello world!')
```

三、实验

1、主要功能实现

```
 5 def print_regrex_object(regrex):
 6     if regrex:
 7         print "\tregrex span:", regrex.span()
 8         print "\tregrex span:", regrex.span(0) # same as regrex.span()
 9         try:
10             print "\tregrex span_2:", regrex.span(2)
11         except:
12             print "\tNOT SUPPORT"
13 
14         print "\tregrex start end:", regrex.start(), regrex.end()
15         try:
16             print "\tregrex start_1 end_1:", regrex.start(1), regrex.end(1)
17         except:
18             print "\tNOT SUPPORT"
19 
20         print "\tregrex group:", regrex.group()
21         print "\tregrex group_0:", regrex.group(0) # same as regrex.group()
22         try:
23             print "\tregrex group_1: ", regrex.group(1)
24         except:
25             print "\tNOT SUPPORT"
26         try:
27             print "\tregrex group_2: ", regrex.group(2)
28         except:
29             print "\tNOT SUPPORT"
30         try:
31             print "\tregrex group_3: ", regrex.group(3)
32         except:
33             print "\tNOT SUPPORT"
34         try:
35             print "\tregrex group_2_3: ", regrex.group(2,3)
36         except:
37             print "\tNOT SUPPORT"
38     else:
39         print "\tNOT MATCHED"
```

```
42 def match_and_search(pattern, str):
43     print "TEST CASE: pattern :", pattern , "\tstr:", str
44     print "\tMATCH"
45     regrex = re.match(pattern, str)
46     print_regrex_object(regrex)
47 
48     print "\tSEARCH"
49     regrex = re.search(pattern, str)
50     print_regrex_object(regrex)
```

2、测试机地址，完整程序

work@nj01-nlp-test01.nj01.baidu.com:/home/work/tianzhiliang/test/python/regex

参考资料

1、只讲python的语法，包含match、search、sub等，有例子、清晰

http://www.runoob.com/python/python-reg-expressions.html

2、比上面详细，包含compile等

http://www.cnblogs.com/huxi/archive/2010/07/04/1771073.html