主要介绍Python中常用的正则表达式处理函数。

## **re.match**

　　re.match 尝试从**字符串的开始**匹配一个模式，如：下面的例子匹配第一个单词。

[IMG_259](http://www.cnblogs.com/sevenyuan/archive/2010/12/06/javascript:void(0);" \o "复制代码)

import re  
text = "JGood is a handsome boy, he is cool, clever, and so on..."  
m = re.match(r"(\w+)\s", text)  
if m:  
print m.group(0), '\n', m.group(1)  
else:  
print 'not match'

[IMG_259](http://www.cnblogs.com/sevenyuan/archive/2010/12/06/javascript:void(0);" \o "复制代码)

re.match的函数原型为：re.match(pattern, string, flags)

第一个参数是正则表达式，这里为"(\w+)\s"，如果匹配成功，则返回一个Match，否则返回一个None；

第二个参数表示要匹配的字符串；

第三个参数是标致位，用于控制正则表达式的匹配方式，如：是否区分大小写，多行匹配等等。

## **re.search**

　　re.search函数会在字符串内查找模式匹配,只到找到第一个匹配然后返回，如果字符串没有匹配，则返回None。

[IMG_259](http://www.cnblogs.com/sevenyuan/archive/2010/12/06/javascript:void(0);" \o "复制代码)

import re  
text = "JGood is a handsome boy, he is cool, clever, and so on..."  
m = re.search(r'\shan(ds)ome\s', text)  
if m:  
print m.group(0), m.group(1)  
else:  
print 'not search'

[IMG_259](http://www.cnblogs.com/sevenyuan/archive/2010/12/06/javascript:void(0);" \o "复制代码)

re.search的函数原型为： re.search(pattern, string, flags)

每个参数的含意与re.match一样。

**re.match与re.search的区别：**re.match只匹配字符串的开始，如果字符串开始不符合正则表达式，则匹配失败，函数返回None；而re.search匹配整个字符串，直到找到一个匹配。

## **re.sub**

　　re.sub用于替换字符串中的匹配项。下面一个例子将字符串中的空格 ' ' 替换成 '-' :

import re  
text = "JGood is a handsome boy, he is cool, clever, and so on..."  
print re.sub(r'\s+', '-', text)

 re.sub的函数原型为：re.sub(pattern, repl, string, count)

其中第二个函数是替换后的字符串；本例中为'-'

第四个参数指替换个数。默认为0，表示每个匹配项都替换。

re.sub还允许使用函数对匹配项的替换进行复杂的处理。如：re.sub(r'\s', lambda m: '[' + m.group(0) + ']', text, 0)；将字符串中的空格' '替换为'[ ]'。

## **re.split**

　　可以使用re.split来分割字符串，如：re.split(r'\s+', text)；将字符串按空格分割成一个单词列表。

## **re.findall**

　　re.findall可以获取字符串中所有匹配的字符串。如：re.findall(r'\w\*oo\w\*', text)；获取字符串中，包含'oo'的所有单词。

## **re.compile**

　　可以把正则表达式编译成一个正则表达式对象。可以把那些经常使用的正则表达式编译成正则表达式对象，这样可以提高一定的效率。下面是一个正则表达式对象的一个例子：