Python使用的是pcre正则

###正则组成：

原子，元字符，模式修正符

###原子：

组成正则表达式的最小单位。一个re至少需要一个原子。

原子有两部分组成：

所有可见字符都是原子：a,b,c,d,….A,B,C,D,….你，我，他，+ -\*、只要能看到的都是

所有不可见字符也都是原子：空格，\t缩进，\n换行，\r,,,,都是原子。

特殊定义原子（专门给正则表达式设计的，转义字符，共7个）：

\d 表示0~9之间任意一个字符 （d：digit）

\D 表示出了0~9以外的任意一个字符

\s 表示任意一个空白字符（s: space）

\S 表示任意一个非空白字符（即可见字符）

\w 表示0~9，a~z，A~Z和\_（下划线）中的任意一个字符

\W 表示除了0~9，a~z，A~Z和下划线\_ 以外的任意一个字符

. 表示任意一个字符除了\n （因为\n是回车换行）

我的python无法识别“.”

###python中的Re模块：正则表达式的专用模块，提供re相关的操作方法

Class 1.py查看详情

Re运行原理：将红框百度放到要查询的字符串中从头到尾一次匹配，直到匹配到。可通

过代码要求re在匹配到第一个符合的结果后继续向后查找有没有匹配的



*import* re  
"""  
###search方法：朝招符合正则表达式的内容  
  
##准备正则表达式,python中使用r方式写  
 ## pattern = "baidu"  
 # #所以，要写成r模式的正则  
  
pattern = r"baidu"  
  
##准备查找的字符串  
url = "http://www.baidu.com"  
  
##匹配操作  
 ##匹配不到返回None  
result = re.search(pattern,url)  
print(result)  
 ##返回：<re.Match object; span=(11, 16), match='baidu'>  
 ##11,16表示“baidu这个字符串在url中的位置是第11到第16个字符，  
"""  
  
###原子测试：  
##准备正则表达式  
pattern = r"\d+"  
  
##准备查找的字符串  
url = "2664564315"  
  
##匹配操作  
##search是用来查找符合的第一个字符  
result = re.search(pattern,url)  
print(result)  
"""  
返回：None，原因，如果用r"\d"在url中查找，他会逐一判断url中的每一个字符是不是0~9的数字，若不是，则返回none  
但如果我们在url中加入一个数字2  
返回：<re.Match object; span=(14, 15), match='2'>  
若pattern=r"\D"即，查找除了0~9以外的字符，则第一个h就被找到  
url被改为13254564（全是数字）则用\D返回的就是None，因为url全是数字  
但如果在url=13254\*564，用\D就能找到\*  
url = 13254 564,\D可以查找出空格。  
\s可以查找到所有不可见字符  
url = "15364564\t315"  
 <re.Match object; span=(8, 9), match='\t'>  
url = "15364564\n315"  
 <re.Match object; span=(8, 9), match='\n'>  
\w表示找到0~9，a~z,A~Z,\_，，，，  
url = "15364564315"  
 <re.Match object; span=(0, 1), match='1'>  
url = "\ 2664564315"  
 <re.Match object; span=(2, 3), match='2'>  
\W是小w的所有非集  
url = "+2664564315"  
 <re.Match object; span=(0, 1), match='+'>  
. 表示任意字符  
拓展：  
r"\d+\"表示找到第一个数字后，取它本身和后面所有数字  
url = "2664564315"  
 <re.Match object; span=(0, 10), match='2664564315'>  
"""

###元字符：（原子修饰符）

##需要0~9中奇数组成的原子：1,3,5,7,9

这时候只用\d就不够用了