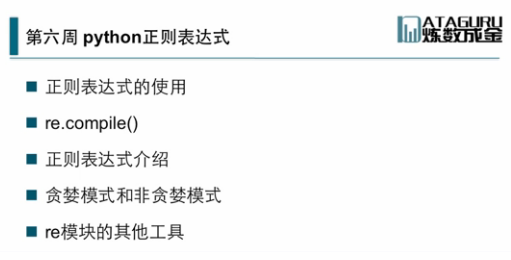
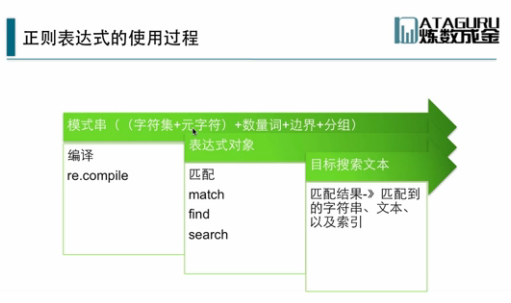
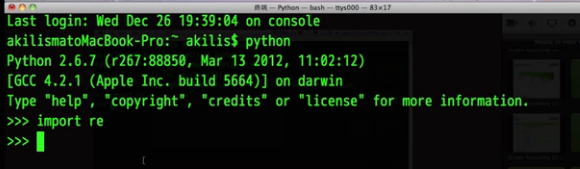
正则表达式





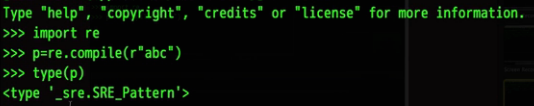
编译模式串 –> 匹配表达式对象 ->目标搜索文本  
先使用一下re模块



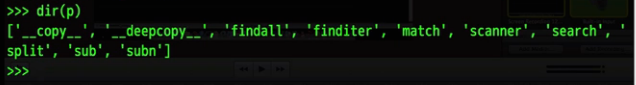
了解先编译模式字符串 到 匹配表达式对象 到最后的目标搜索文本



r表示raw 也就是 r”abc” 对字符串中出现的转义字符不进行转义

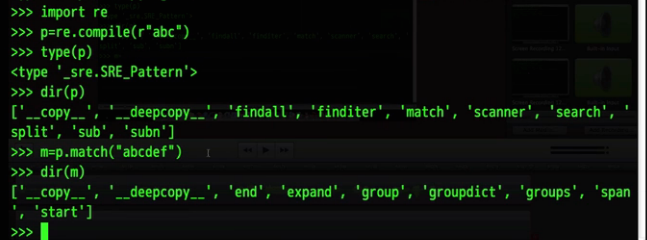


看一下p下面有哪些东西



对于编译好的模式对象 有很多可疑使用的方法

现在如何从一个目标串里面 查找

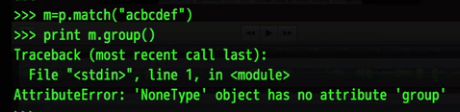


可疑看到匹配目标对象之后 有哪些方法

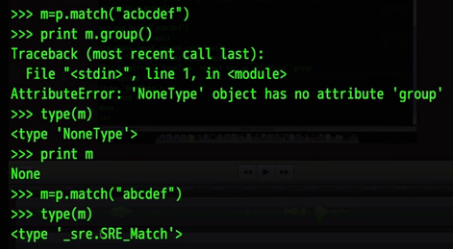


所以 在字符串abcdef中是可以找到模式串

现在修改以下是



告诉我们这个m对象是没有这个东西的 没有匹配到的话 是一个None值



这个图就对比了 一个是可以匹配上 一个是不能匹配上

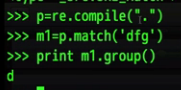
如果找到是一个RE对象 否则是None值



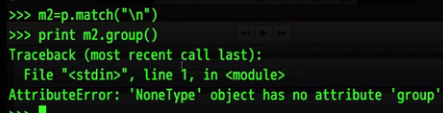
元字符 表示一些通配符

.表示出了换行符之外的任意字符

测试一下



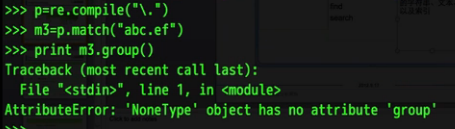
可以看到 这个.匹配了第一个字符d



也就是m2就是没有匹配到换行符 .就表示了除了换行符以外的其他字符

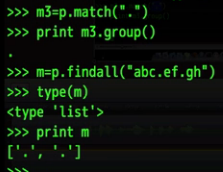
现在看转义字符

如果我想匹配原生的. 那就应该写成\.

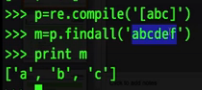


没有找到匹配的这样一个字符串的

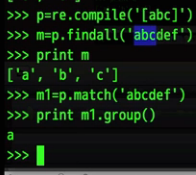
在我这个模式字符串中 是没有这个匹配规则的



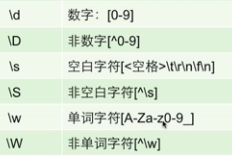




对于match操作

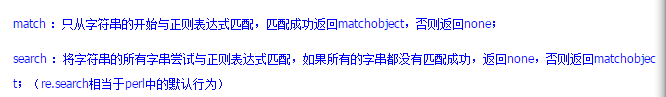


Findall是抓出所有的结果 match是抓出第一个结果[个人认为是否是 第一位的结果匹配]



===== 网上找的

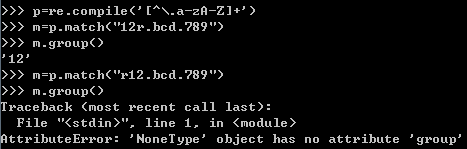




findall是search发扬光大版

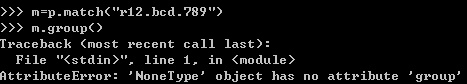
search仅仅返回匹配的结果 但是 findall会返回所有的结果

正则就是 不能使. 和 字母



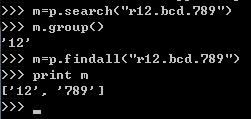
这个说明了

Match仅仅是匹配字符串的开头部分 找不到 就完蛋了





这个说明了 search()不同于match()的地方 就是 search不仅仅局限于字符串的开头 找到就立刻返回



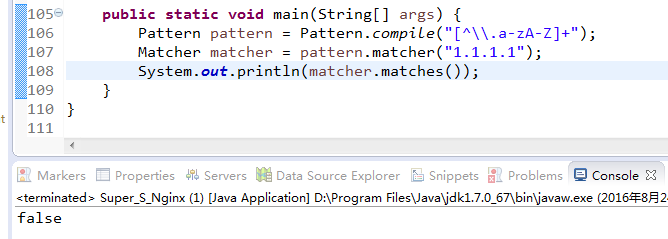
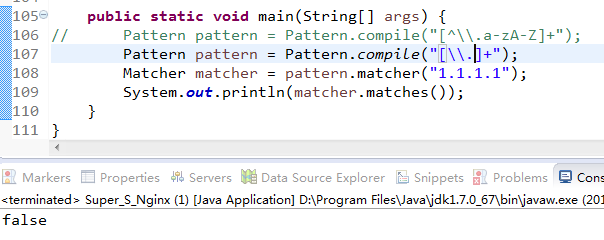
这个说明了findall是找所有匹配的子部分 但是search()找到第一个

=== 个人理解：

语义：match 匹配 在Python中理解为 整个一打眼看是不是符合 整体匹配 那么 如果一个String开始就不匹配 那么 就谈不上整体的符合 所以 match和search会存在上面的区别

但是 search的字面意思是查找 查找就找到就行 不局限于字符串开头必须相似

===== java中和python的区别



注意 python的match() 但是 java中的是matcher()

同事 python的p可以调用match() findall() search() 返回的结果类型不一样

所以 python中match()这一步骤不是必要的

但是 java中要进行正则匹配 必须有Matcher对象 这个必须由pattern对象的matcher方法生成

所以 java中的matcher()方法是必要的

语义上理解：

Python中 match或者search 等 是正则本身是主动 到 目标字符串中做匹配

Java中 是反过来 目标字符串 去匹配正则表达式的模式