1.1什么是正则表达式？

1. 用某种模式去匹配一类字符串的公式，主要用来描述字符串匹配的工具
2. 描述了一种字符串匹配的模式。用来检查字符串中是否含有某种字串、将匹配的字串做替换或者从某个字符串中取出符合某个条件的字串
3. 它由普通字符（a~z）和特殊字符（元字符）组成的文字模式
4. 它是记录文本规则的代码
5. 用一个字符串描述一个特征，然后去验证另一个字符串是否符合这个特征

匹配

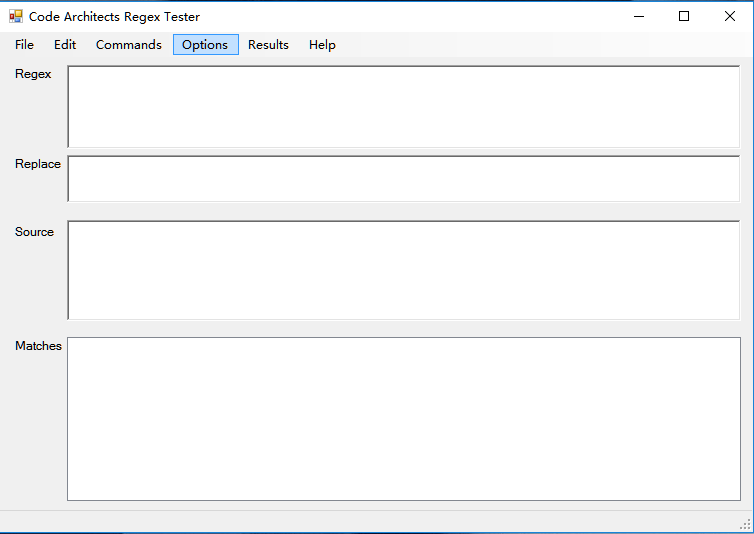
1. 形容词性的匹配，一个字符串匹配一个正则表达式
2. 动词性的匹配，在文本或字符串里匹配正则表达式
3. 名词性匹配，字符串中满足正则表达式的一部分

正则表达式应用

1. 验证字符串
2. 查找~
3. 替换~
4. 提取~

1.2测试工具

Code Architects Regex Tester



Regex区域：书写正则表达式

Replace：书写替换相关的表达式

Source：被匹配的字符串或文本

1.3理解元字符

两类：匹配位置和匹配字符

匹配位置（3个）：^、$、\b

^（脱字符号）和$字匹配一个位置，分别为匹配行的开始和结尾

\b可以匹配单词的开始或结尾

如：^String$

匹配以“String”开头和结尾的字符串

匹配字符（7个）：.、\w、\W、\s、\S、\d、\D

.（点号）：匹配除换行外的任意字符

\w：匹配单词字符（字母，数字，下画线和汉字）

\W：匹配任意的非单词字符

\s：匹配任意的空白字符（如空格、制表符、换行符等）

\S：匹配任意的非空白字符

\d：匹配任意数字

\D：匹配任意的非数字

如：\ba\w\w\w\w\b

匹配以’a’开头的长度等于8的任意单词

1.4文字匹配

字符类：一个字符集合，如果该字符集合中任意一个字符被匹配，它就会找到该匹配项。在[]中定义

如：[0123456]

可以匹配数字0123456中任何一个

注：连接符“-”定义字符范围，上式可写为[0-6]

字符“^”表示否定该字符,匹配除该字符以外的其他字符，如[^0123456]

字符转义：匹配在正则表达式中定义的一些特殊元字符，如^。如需匹配，使用转义字符“\”加在前面

\t：制表符

\r：回车符

\f：换页符

\n：换行符

限定符：匹配多个字符时使用，指定特定字符或字符集自身重复出现的次数

{n}：重复n次

{n,}：重复至少n次

{n,m}：重复至少n次，至多m次

\*：重复至少0次

+：重复至少1次

?：重复0次或1次

如：\d{3,4}

匹配3位或4位整数

注：若在前面的限定符后加“?”，表示尽可能少地重复字符“?”前限定符号的次数，这称为懒惰匹配。反正为贪婪匹配

例：a.\*b 为贪婪匹配，匹配以a开头，b结尾的最长字符串

a.\*?b 为懒惰匹配，匹配以a开头，b结尾的最短字符串

1.5替换

同时满足多个需求，使用替换，用“|”表示（感觉像或）

如：Jack|jack等同于[Jj]ack可匹配两种字符串

分组：又称子表达式，可把一个正则表达式分成多个组。用字符“(”和“)”

例：（\d{1,3}\.）{3}

将子表达式\d{1,3}\.匹配的字符串重复3次，如“12.34.56”

反向引用

一个正则表达式分组之后，每一个组将自动被赋予一个组号。从左到右，第一组号为1。

反向引用提供查找重复字符组的简便方法，再次匹配相同字符

\数字 使用数字命名的反向引用

\k<name> 使用指定命名的反向引用

例：\b(\w)(\b)\1\2\b

匹配单词的开头出，然后匹配一个字符和数字，在重复匹配该字符和数字，最后是单词结尾处

1.6其他运算

零宽度断言（断言）：匹配一个位置，且这个位置满足一定条件

如：^

负向零宽度断言：匹配不止一个位置，就是“反义”

匹配选项：如忽略大小写、处理多行等

注释格式：

（?#注释）