

# 正则表达式基础知识和在线测试工具

《用 Python 玩转数据》课程资料 by Dazhuang@NJU

例如要在一篇文档中寻找字符串 `he`，则可以使用正则表达式 `he`，它可以匹配出字符串 `he`，如果忽略大小写的话还可以匹配出 `He`、`HE` 这样的字符串，这是最简单的正则表达式。但这种方式除了匹配出单独的 `he` 外还会匹配例如“`hello`”中的 `he`。如果只想找出字符串 `he`，可以使用元字符“`\b`”，它匹配单词的开始或结尾，即单词边界，因此可用“`\bhe\b`”匹配字符串 `he`。再例如正则表达式“`\d{1,3}`”可以匹配包含 1 到 3 个数字字符的字符串如“`11`”和“`222`”。另外还可以利用括号(`exp`)指定子表达式 `exp`（也称为分组），这样不仅可以重复单个字符也可以重复多个字符构成的子表达式。

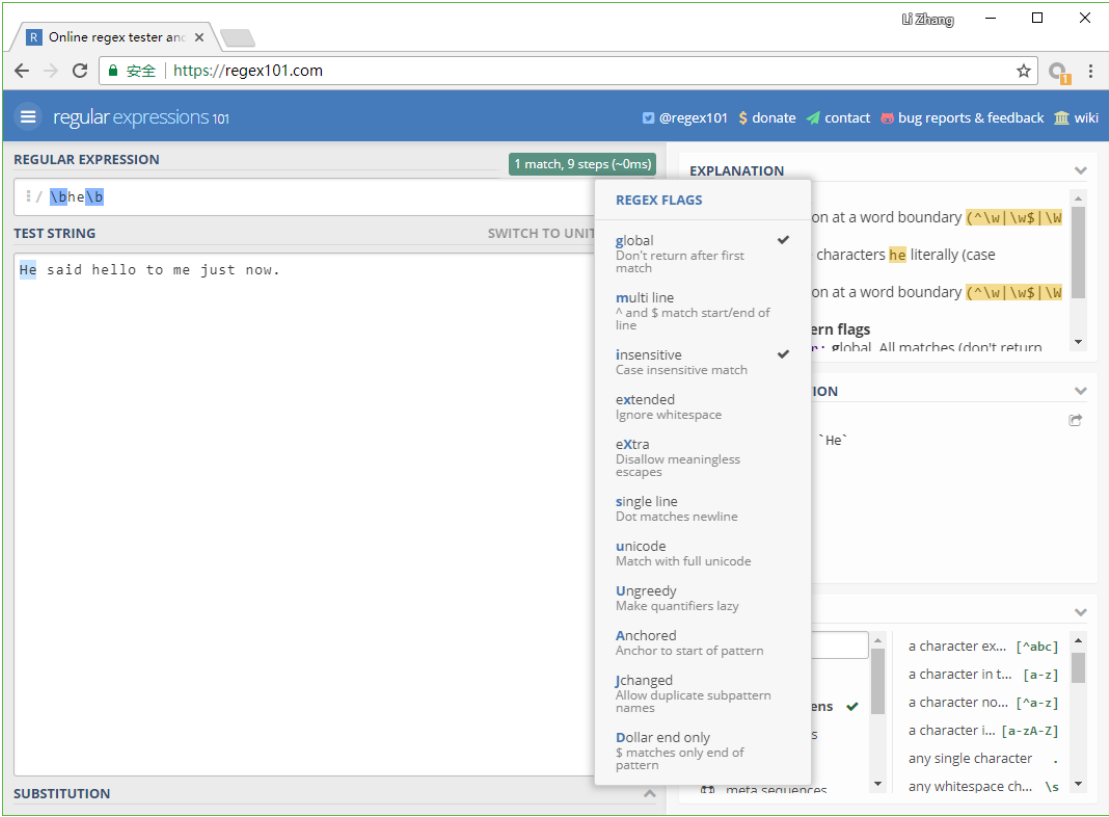
正则表达式中有很多元字符，常用的元字符如下表所示：

表 1 正则表达式常用元字符

元字符	描述
.	匹配除换行符外的任意字符
*	重复前面的子表达式 0 次或多次
+	重复前面的子表达式 1 次或更多次
?	重复前面的子表达式 0 次或 1 次
^	匹配字符串的开始
\$	匹配字符串的结束
{n}	重复 n 次
{n,}	重复 n 次或更多次
{n,m}	重复 n 到 m 次
\b	匹配单词的开始或结尾即单词边界，“\B”匹配非单词边界
\d	匹配数字，“\D”匹配任意非数字字符
\s	匹配任意空白符，“\S”匹配任意非空白符
\w	匹配任意字母、数字或下划线的标识符字符，“\W”匹配任意非标识符字符
[a-z]	匹配指定范围内的任意字符
[^a-z]	匹配任何不在指定范围内的任意字符

特别的，当正则表达式中包含重复的限定符如“`*`”时，通常会匹配尽可能多的字符，例如对于正则表达式“`a.*b`”，它会匹配以 `a` 开头以 `b` 结尾的最长字符串，如果用它来搜索 `aabbab` 时，它会匹配整个字符串 `aabbab`，这种方式称为贪婪匹配。如果想匹配尽可能少的字符，即进行懒惰匹配，则只要在“`.*`”后加上“`?`”构成“`.*?`”，例如用懒惰匹配来搜索 `aabbab` 时，会匹配 `aab` 和 `ab`。

正则表达式常常比较复杂，所以可利用正则表达式在线测试/调试工具帮助正则表达式的书写。下图所示为著名的实现此功能的网站（<https://regex101.com/>），可设置忽略大小写（insensitive 选项）等选项。



对正则表达式有兴趣的学习者可以继续深入研究。