

正则表达式简明教程

正则是一种字符串匹配模式，在处理文本时很有用。最常见的操作就是用于查找和替换。

说到处理文本，其实我们每天敲的代码就是文本，因此常用的代码编辑器的查找替换工具基本都支持正则语法的。



先说明一下，接下来的内容都以《We will rock you》的歌词测试文本。

► 点击展开歌词

使用的正则测试工具是 [Regex 101](#)。

这里建议读者看的过程中，同时打开该网站，把歌词贴进去，每个案例都验证一遍。也建议稍微改动一下正则，看看匹配结果仍是否与自己的理解一致。跟着动手，学习效果要好一些。

1. 精确匹配

正则是用来描述字符串的一种模式（pattern），或者说规律。最平凡的用法，就是精确查找。比如我要找到歌词中的所有“the”。正则写成 the 即可。

REGULAR EXPRESSION

1 match, 4 steps (~0ms)

/ the

/

TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will rock you
Singing it now
We will, we will rock you

@稀土掘金技术社区

上图只找到了一个 the，而不是所有的。这是因为正则本身是分两部分的，一部分是模式，另一部分是修饰符（flags，或者叫标志位）。一个常用的修饰符是 g，它单词 global 的简写，表示全局查找。

REGULAR EXPRESSION

5 matches, 38 steps (~0ms)

/ the

/ g

TEST STRING

SWITCH TO UNIT TESTS ▸

```
Buddy, you're a boy make a big noise  
Playing in the streets gonna be a big man someday  
You got mud on your face  
You big disgrace  
Kicking your can all over the place  
Singing  
We will, we will rock you  
We will, we will rock you  
Buddy you're a young man, hard man  
Shouting in the street gonna take on the world someday  
You got blood on your face  
You big disgrace  
Waving your banner all over the place  
We will, we will rock you  
Singing it now  
We will, we will rock you
```

@稀土掘金技术社区

此时，我们找到了所有“the”。接着我们再找所有“we”。

REGULAR EXPRESSION

4 matches, 30 steps (~0ms)

/ we

/ g

TEST STRING

SWITCH TO UNIT TESTS ▶

```
Buddy, you're a boy make a big noise  
Playing in the streets gonna be a big man someday  
You got mud on your face  
You big disgrace  
Kicking your can all over the place  
Singing  
We will, we will rock you  
We will, we will rock you  
Buddy you're a young man, hard man  
Shouting in the street gonna take on the world someday  
You got blood on your face  
You big disgrace  
Waving your banner all over the place  
We will, we will rock you  
Singing it now  
We will, we will rock you
```

@稀土掘金技术社区

然而，同时我们也希望找到文本中“We”，w 字符是大写的。此时可以用另外常见的标识符 i，单词 ignoreCase 或者 insensitive 的首字母，表示忽略大小写。

REGULAR EXPRESSION

8 matches, 44 steps (~1ms)

/ we

/ gi

TEST STRING

SWITCH TO UNIT TESTS ▶

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will rock you
Singing it now
We will, we will rock you

@稀土掘金技术社区

无论 the 或 we，这种模式匹配都是精确匹配，如果正则只是输入什么就查找什么，那么其存在的意义就没有那么大。而它的强大之处在于能实现模糊匹配。

2. 横向模糊匹配

比如我们想找到歌词中所有连续出现的“e”。

REGULAR EXPRESSION

32 matches, 65 steps (~0ms)

/ e{1,2}

/ g

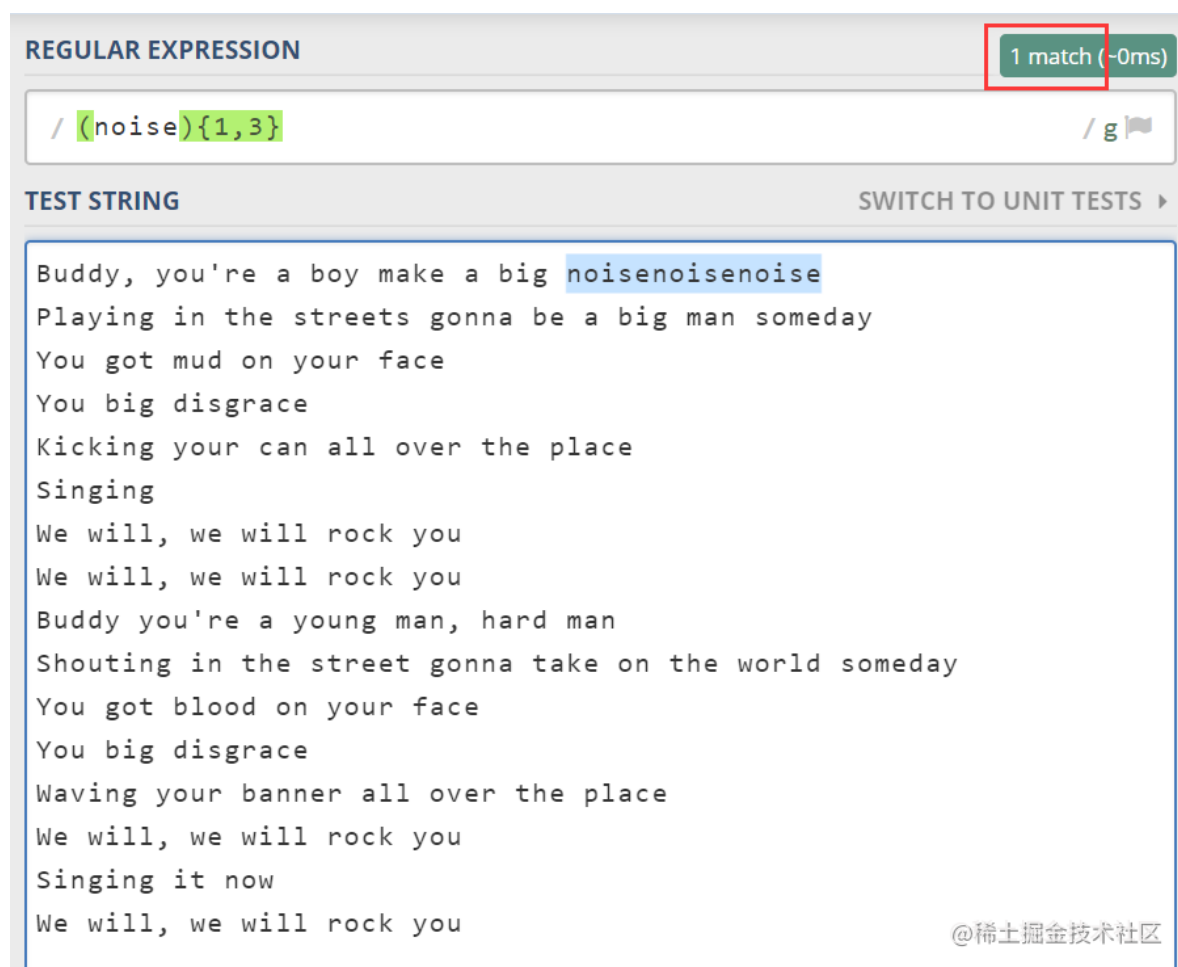
TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will rock you
Singing it now
We will, we will rock you

@稀土掘金技术社区

图中正则形如 $p\{m,n\}$ ，表示 p 至少连续出现 m 到 n 次（包括 m 、 n ）。 p 可以是一个子模式，不一定只是一个字符。



上图中，为了测试我修改了部分歌词。其中正则使用了括号，括号如你所料一样，起到了高优先级的作用。表示 noise 这个整体重复出现了 1 到 3 次。

不知道此时你是否有疑问，{1,3} 表示 1 到 3 次。为啥上面的匹配结果只有一个呢？而不是匹配到 3 个 noise。又或者 noisenoise 和 noise，这两个结果呢？

这是因为量词有贪婪和惰性之分。{1,3} 这个量词是贪婪的，能满足条件的话，它会尽可能多地匹配。可以在量词的后面加个问号，让其变为惰性的。

REGULAR EXPRESSION

3 matches (~1ms)

/ (noise){1,3}? / g

TEST STRING

SWITCH TO UNIT TESTS ▶

Buddy, you're a boy make a big noisenoisenoise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will rock you
Singing it now
We will, we will rock you

@稀土掘金技术社区

确实够懒得的，找到一个就满足了。量词后面的这个问号，仿佛是在问量词，“可以别再贪了吗？”

量词的含义清楚了，下面我们来看一些简写形式。

- `*` 等价于 `{0,}`。即任意多个。
- `+` 等价于 `{1,}`。即至少一个
- `?` 等价于 `{0,1}`。即有一个或者没有
- `{m}` 等价于 `{m,m}`

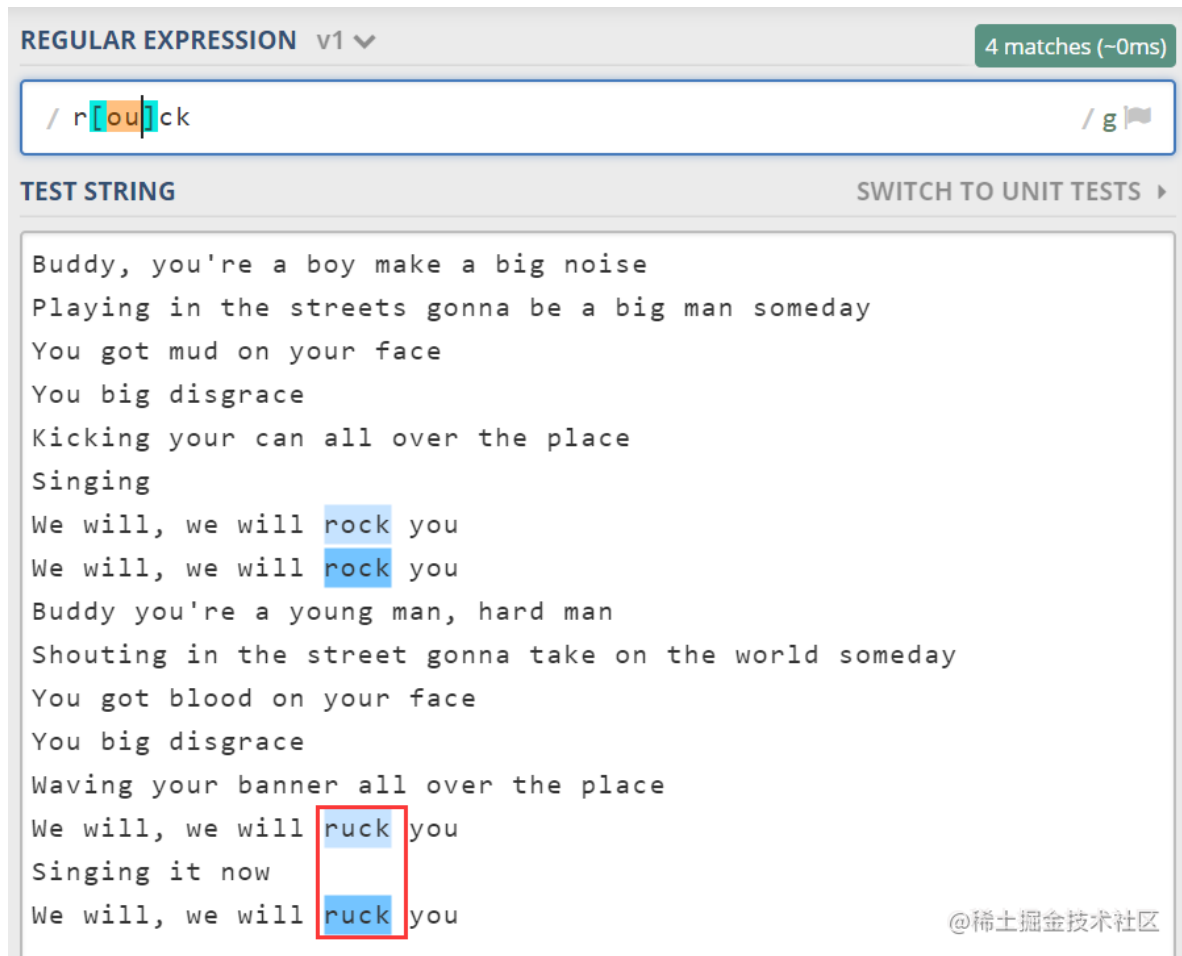
这里要说明的是？这时就可能两个含义。即一个表示惰性模式，一个表示量词。

其实二者很好区分，在量词之后的？才表示惰性匹配。比如正则 `bo??y`，第一个问号表示量词 `{0,1}`，第二个表示量词是惰性的。

量词的存在，能让正则可以模糊匹配，即很少的模式代码就能匹配一长串。我称之为横向模糊匹配。还有一种纵向的模糊匹配。

3. 纵向模糊匹配

假设歌词中有几处不小心把“rock”写成“ruck”。我们需要找到二者，可以使用字符集 `r[ou]ck`。效果如下：



其中 `[ou]`，这种方括号括起来的模式就是字符集。它是一个集合，匹配“o”或者“u”。又比如我们要找到所有 a 到 e 的字符，可以写成 `[abcde]`。这种连续的字符也可以简写成 `[a-e]`。

REGULAR EXPRESSION v1 92 matches (~1ms)

/ [a-e]

TEST STRING SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

字符集是集合的意思，而集合有补集。正则里在方括号内开头加上脱字符，来表示取反^[^a-e]，匹配一个不是 a、b、c、d、e 的某字符。

REGULAR EXPRESSION v1

371 matches (~1ms)

/ `[^a-e]`

/ g

TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

字符类的含义搞清楚了，下面我们来看一下常见的简写形式

- `\d` 等价于 `[0-9]`。表示是一位数字。digit 的首字母。
- `\D` 等价于 `[^0-9]`。
- `\w` 等价于 `[0-9a-zA-Z_]`。表示数字、大小写字母和下划线。word 的首字母，也称单词字符。
- `\W` 等价于 `[^0-9a-zA-Z_]`。
- `\s` 等价于 `[\t\v\n\r\f]`。表示空白符，包括空格、水平制表符、垂直制表符、换行符、回车符、换页符。记忆方式：s 是 space character 的首字母。
- `\S` 等价于 `[^\t\v\n\r\f]`。
- `.` 等价于 `[^\n\r\u2028\u2029]`。点是通配符，表示几乎任意字符。

字符集是正则实现模糊匹配的另外一种方式，具体到某一位上，要匹配的字符可以是不确定的，我称之为纵向模糊匹配。

量词和字符组掌握了的话，基本上正则问题能解决一多半。这里再举一个例子。找到所有以“ing”结尾的单词。

REGULAR EXPRESSION v1 6 matches (~0ms)

/ \w*ing / g

TEST STRING SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

上面使用的是贪婪量词，如果使用惰性量词的话，情形会有所不同。

REGULAR EXPRESSION v1 8 matches (~0ms)

/ \w*?ing / g

TEST STRING SWITCH TO UNIT TESTS

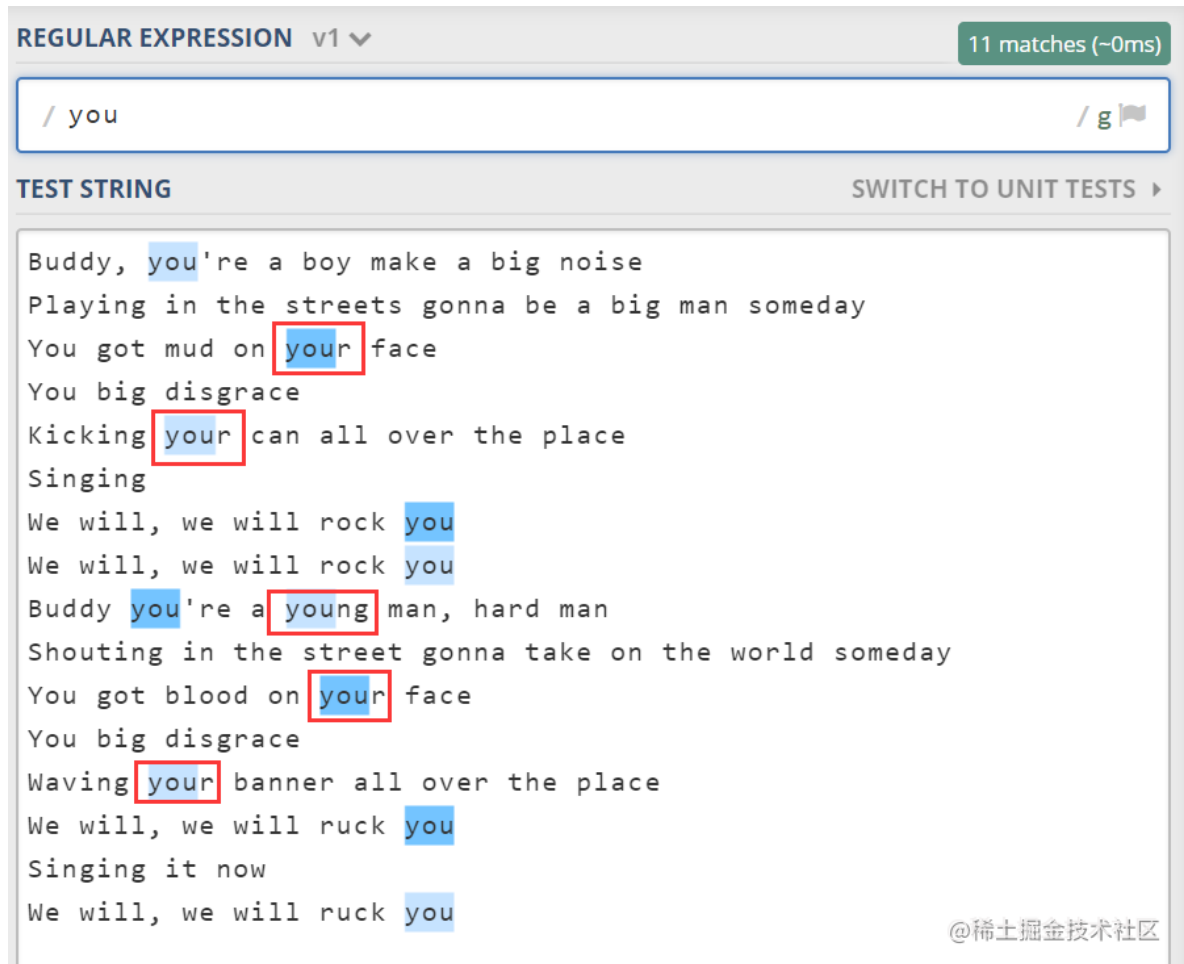
Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

此时，“singing”这个单词分成了“sing”和“ing”。要完整的匹配一个单词。需要匹配位置。

4. 匹配一个具体位置

比如匹配“you”这个单词，可能会匹配到“your”中的 you。



此时我们可以使用`\b`。b 是单词 boundary 的首字母。它表示匹配一个位置，这个位置某一边是`\w`，另一边是`\W`。也就是一边是单词字符，一边是非单词字符，因此它叫单词边界。

如果对“位置”这一概念，理解得还是不太透彻，我们可以具体看看 `\b` 到底长什么样。

REGULAR EXPRESSION v1 ▾194 matches (~1ms)

/ \b/ g

TEST STRINGSWITCH TO UNIT TESTS ▸

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

注意上图中得粉色虚线。它们就是一个一个位置。请看看每一个是不是两边一个是单词字符，另一个是非单词字符。

位置也是有反义的。比如 \B 表示非单词边界。我们也可以看看。

REGULAR EXPRESSION v1 ▾ 270 matches (~0ms)

/ \B / g

TEST STRING SWITCH TO UNIT TESTS ▸

```
Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you
```

@稀土掘金技术社区

有了单词字符后，要准确的匹配单词“you”，可以使用\byou\b。

REGULAR EXPRESSION v1 ▾ 6 matches (~0ms)

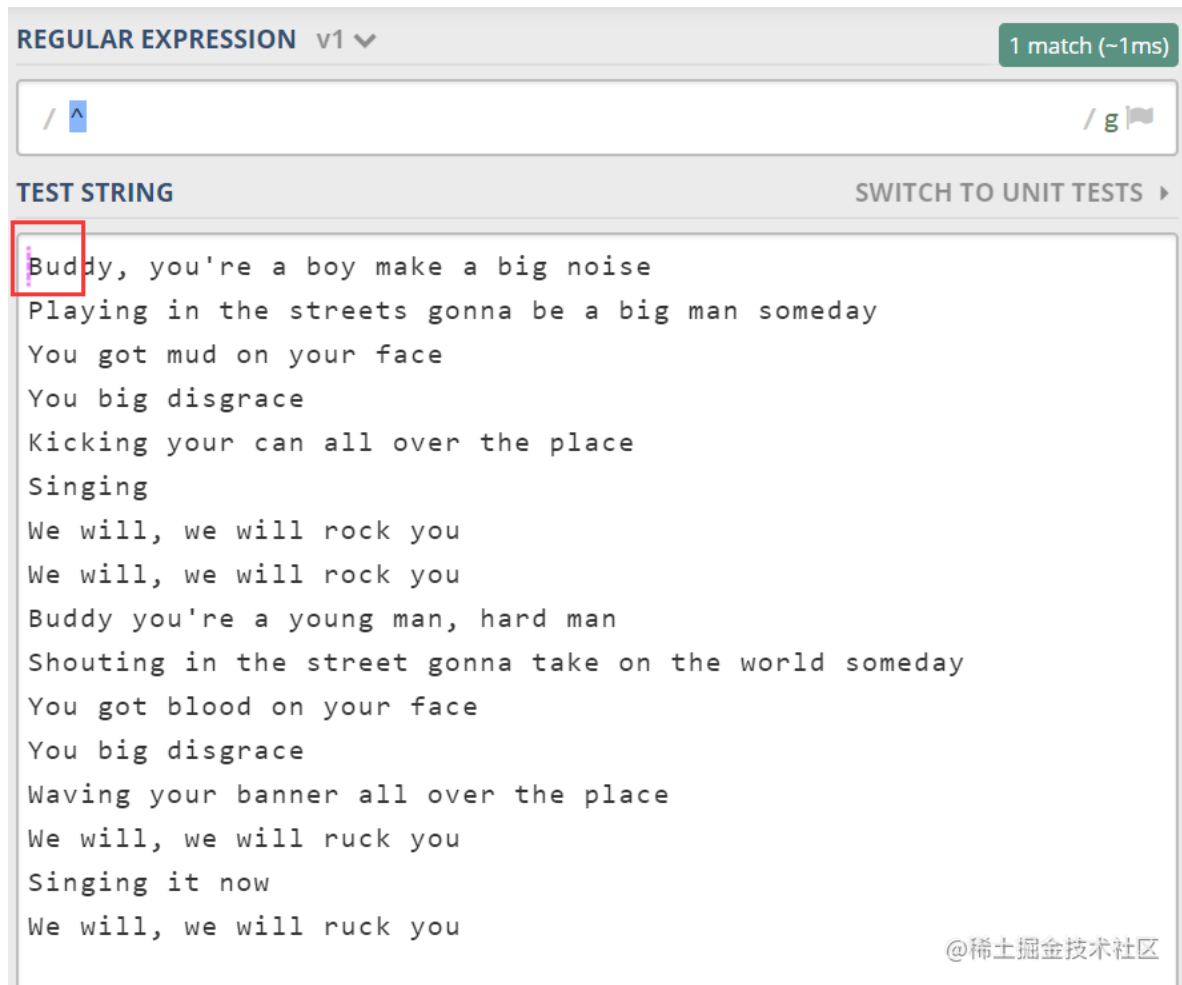
/ \byou\b / g

TEST STRING SWITCH TO UNIT TESTS ▸

```
Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you
```

@稀土掘金技术社区

除了单词边界这种位置之外，估计大家应该知道 ^ 和 \$。它匹配整个文本的开头和结尾。



The screenshot shows a web-based regular expression testing interface. At the top, it says "REGULAR EXPRESSION v1" and "1 match (~1ms)". The input field contains the regex "^" and the flags "g" and "i". Below the input field, there's a "TEST STRING" section with a text area containing the lyrics of "Smells Like Teen Spirit". The first character of the text, "B", is highlighted with a red box, indicating a match for the "^" regex. The text in the test string is: "Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you". At the bottom right, there is a watermark "@稀土掘金技术社区".

还记得前面我们找“we”吗，如果我们想找到所有行开头的 we 单词。我们可以使用多行模式：

REGULAR EXPRESSION v1

4 matches (~0ms)

/ ^we

/ igm

TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

此时修饰符里多了一个 m，是 multiline 的首字母，表示多行匹配。所谓多行匹配，就是说 ^ 和 \$，可以匹配行开头和行结尾，不再局限于整个文本的开头和结尾。

除了 \b、\B、^、\$ 外，还有一种断言位置。比如 (?=p)，表示模式 p 前面的位置。

REGULAR EXPRESSION v1

11 matches (~0ms)

/ (?=s) /igm

TEST STRING SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

(?!p)是其反义。还有反向的断言，例如 (?<=p)，表示模式 p 后面的位置。或者说该位置的后面是 p。它也有反义的形式 (?<!=p)。请读者自己尝试看看都匹配了啥。

关于位置这一块儿，多说几句。假如我想找到这样的位置，该位置不能是开头，并且后面的字符是 s，此时该怎么做呢？

REGULAR EXPRESSION v1

8 matches (~1ms)

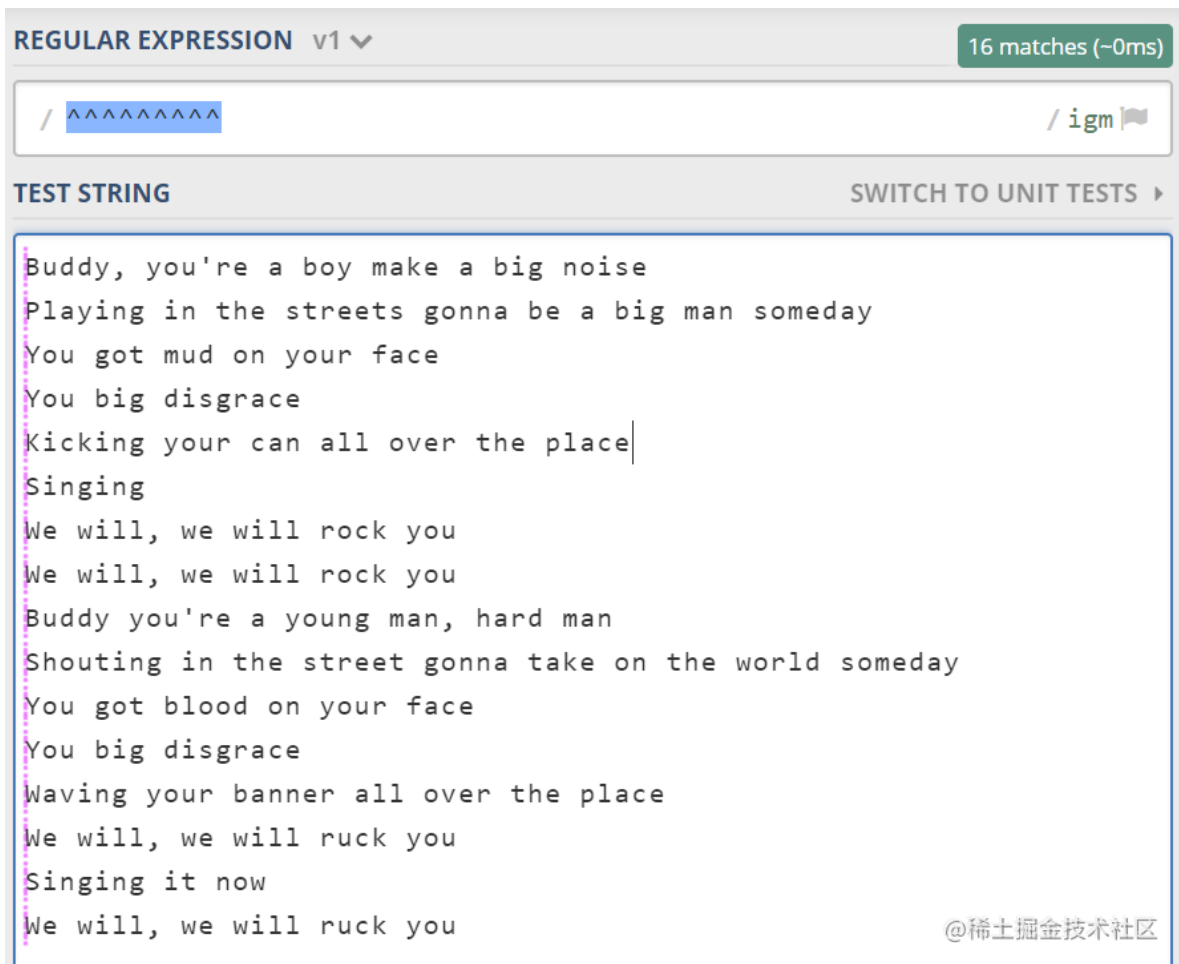
/ (?!\^)(?=s) /igm

TEST STRING SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

(?!^) 其实就是 ^ 的反义。连续写多个位置是没有关系的。比如写 ^^^^。



需要注意的位置不同于字符，是不占地方的，如果说是字符也可以，它则是空字符，没有实际宽度的。

正则要么匹配字符，要么匹配位置。主体内容介绍完了，接下来查缺补漏。

5. 引用

street 里有两个 e，而 all 里有连个 l。此时我想找到所有这样的双棒字母，该怎么做呢？直接使用 `.{2}` 是不行的。因为它就是 `..` 的简写形式，表示两个任意字符。并没有要求这两个字符相同。

/.{2}

/igm

TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

此时，就涉及到了反向引用。参考如下写法：

REGULAR EXPRESSION v1

18 matches (~0ms)

/ (.)\1 /igm

TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

\1 是反向引用，表示第一个括号里捕获的数据。那么 \2 呢，表示第二个括号捕获的。

REGULAR EXPRESSION v1

18 matches (~0ms)

/ `((.))\2`

/ `igm`

TEST STRING

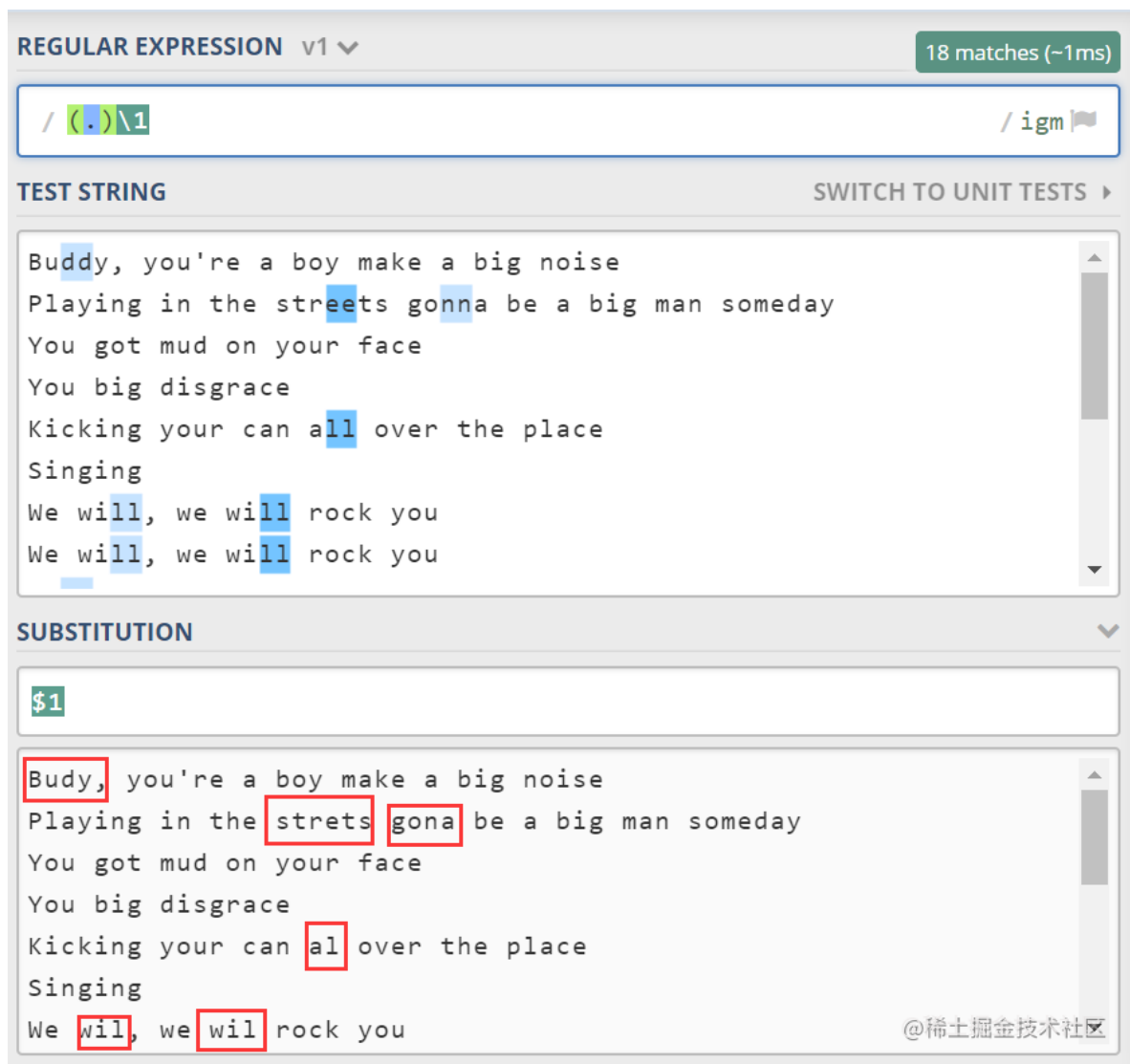
SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

需要注意一点是这里的括号，是平常的括号，而不是像 `(?=p)` 那样特殊语法的括号。

括号捕获的数据，不仅可以在正则里反向引用。也可以配合宿主 API 来使用，外部引用。比如实现滤重：



上面使用了替换，工具内部必然要用到宿主语言相关 API。`$1` 表示外部引用第一个分组捕获的内容。

括号可以用来提供分组功能，又能捕获数据。能否只让括号充当分组功能呢？此时要使用非捕获分组 `(?:p)`，而不是 `(p)`。

6. 分支结构

比如我想找到所有的 `face` 和 `place`。此时该怎么办？

REGULAR EXPRESSION v1

4 matches (~0ms)

/ face|place

/ igm

TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

管道符 |，表示或的关系，多选一。它从左到右面一个个尝试，如果成功，就不再继续尝试了。可以说它是短路的、惰性的。比如用 you|your 去匹配 your 时，它只会匹配到 your 的前 3 个字母。所以分支顺序不同结果可能也会不同。

REGULAR EXPRESSION v1

15 matches (~0ms)

/ you|your

/ igm

TEST STRING

SWITCH TO UNIT TESTS

Buddy, you're a boy make a big noise
Playing in the streets gonna be a big man someday
You got mud on your face
You big disgrace
Kicking your can all over the place
Singing
We will, we will rock you
We will, we will rock you
Buddy you're a young man, hard man
Shouting in the street gonna take on the world someday
You got blood on your face
You big disgrace
Waving your banner all over the place
We will, we will ruck you
Singing it now
We will, we will ruck you

@稀土掘金技术社区

总结

本文，通过案例的形式覆盖了正则常用语法。包括：

- 量词
- 字符集
- 分支结构
- 修饰符
- 位置
- 引用
- 常见简写

如果文中每个例子，你都自己手动输入调试并理解的话，你可以放心地说自己正则已经入门了。