

Vim实战指南(五): 文本替换

Introduction

在编辑代码时, 有时候你希望对变量或者函数重命名。vi提供了强大的替换功能, 比其他普通编辑器的ctrl+r更好用。

替换

vi的替换基于ex命令:

```
:s/old/new/g: 将当前行所有的old替换成new, s是substitute, g是global
:1,10s/old/new/g: 将第1到第10行所有的old替换成new
:%s/old/new/g: 将当前文件所有行的old替换成new, %是第一行的意思
:%s/old/new/gc: 同上, 但是替换前需要确认, c是confirm
:g/pattern/s/old/new/g: 将符合pattern的行内的所有old替换成new, 第一个g是指匹配pattern的所有行, 第二个g是指行内所有的匹配old的字符串
```

正则表达式

下面列出正则表达式中使用的metacharacters:

```
.: 匹配除了换行符之外的任何符号(包含空格), 如a.a匹配aaa和aba
*: 重复前面(零次或多次)匹配符, 如a*匹配a和aa
^: 当出现在表达式前面时, 表示头部匹配, 如^.匹配行首第一个字符
$: 当出现在表达式尾部时, 表示尾部匹配, 如.$匹配行尾第一个字符
\: 转移符, 如*表示匹配*, 而不是重复前面的匹配符
[:]: 匹配中括号中任何一个匹配符, 如[ab]表示匹配a或者b
\\(: 将括号中的模式保存
\\<\\>: 匹配以\\<符号后开始和\\>符号前结束的单词
```

正则表达式可以结合上面的pattern使用，这里不再赘述。

技巧

在程序编写中，你觉得pan有歧义，你希望把所有的pan替换成panic：

```
:%s/pan/panic/g
```

稍后你发现，所有的panic被替换成了panicic，这不是我们想要的。你注意到pan后面都有一个符号，因此你使用：

```
:%s/pan[ ,.:;]/panic[ ,.:;]/g
```

接着你发现，这种匹配仍然有漏网之鱼，如pan=1，仍然需要匹配=号。

事实上你打不用如此麻烦，只需下面简单的替换即可：

```
:%s/\<pan\>/panic/g
```

Summary

本文列出了常用的文本编辑中全局替换的方法。由于正则表达式从来都不是一个简单的话题，本文只是简单的介绍了如何在替换中使用正则表达式，事实上vi支持所有的正则表达式功能，可以完成任何你能想到的替换！

(本文出自csprojectedu.com，转载请注明出处)