

(vim) 搜索与替换

手稿



封面

vim 可以执行搜索和替换，适当地运用这些功能可以极大地提高工作效率。

概述

搜索和替换在文本编辑过程中会经常用到，大部分文本编辑软件都有这两项功能，vim 也不例外。

vim 支持利用正则表达式匹配字符串，支持向前 / 向后搜索，支持匹配项高亮显示，还支持增量搜索。对于一些简单的文本替换工作，可以使用 `r` 或 `R` 命令；如果有更复杂的替换任务，可以使用 vim 的 `:s` 命令。

单行字符搜索

`vim` 支持在一行中搜索单个字符，举例来说，如果要搜索字符 `x`，那么：

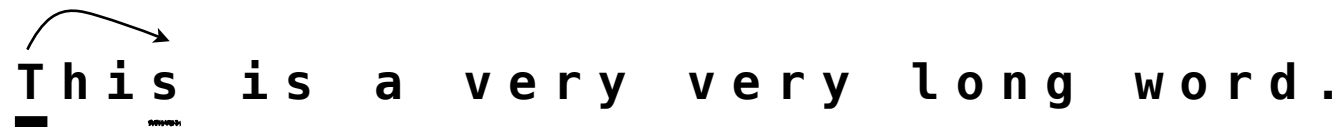
`fx` 当前位置向右查找，将光标移至该行中下一个 `x` 出现的位置

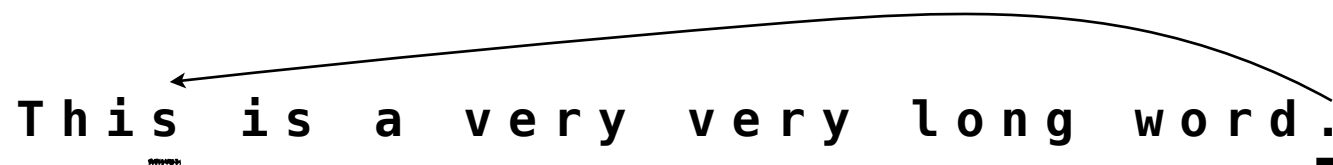
`Fx` 当前位置向左查找，将光标移至该行中前一个 `x` 出现的位置

`tx` 当前位置向右查找，将光标移至该行中下一个 `x` 出现位置的左侧字符

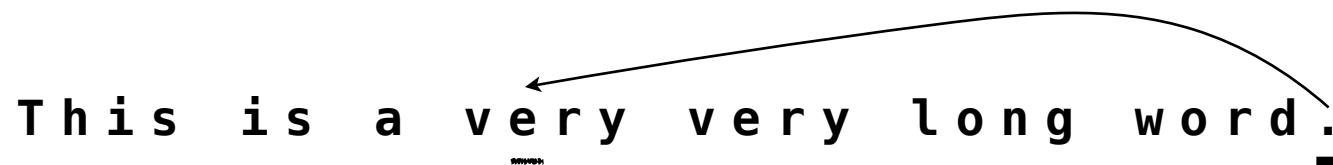
`Tx` 当前位置向左查找，将光标移至该行中下一个 `x` 出现位置的右侧字符

注：在 `f`、`F`、`t`、`T` 命令之后，按下 `;` 会重复这个命令，这些命令前如果有大于零的数字 `n`，则重复 `n` 次。

 `fs`

 `2Fs`

 `tv`

 `2Tv`

向前搜索命令 /

`/pattern` 从光标所在位置开始向后搜索与匹配模式 `pattern` 匹配的字符串。`vim` 找到匹配项时，光标移至匹配项的第一个字符。如果设置了选项 `hlsearch`，则匹配项高亮显示。

找到匹配项后，按下 `n` 键继续搜索下一个，按 `N` 反向搜索上一个。

注：`/`（包括 `?`）命令最近一次使用的匹配模式会被保存在 `"/` 缓冲区。

```
89 /*
90
91 Function linear() is simple, it is used to calculate general linear result.
92
93 linear() conforms to the folloing formula:
94     y = f(x) = a + bx
95 there are three parameters in function linear() which named as a, b and x
96 corresponding to the above formula.
97
98 Among these three parameters, a is the constant term or the y intercept,
99 b is the coefficient of the independent variable, x is independent variable.
100 y or f(x) is dependent variable, and is returned as return value.
101
102 */
103 double linear(double a, double b, double x)
104 {
105     return a + b * x;
106 }
107
108 int main(int argc, char *argv[])
→ /linear
```

特殊字符

在 `vim` 中使用搜索功能时，要注意匹配模式中出现的特殊字符 (`. * [ ] ^ % / \ ? ~ $`)。

- ✦ `.` 匹配任意字符 `/v.ry -> very`
- ✦ `*` 匹配零或多次 `/lo*ng -> loooooooooooooooooong`
- ✦ (使用 `\+` 可以避免零次匹配) `/wo\+rd -> word` (不匹配 `wrd`)
- ✦ (使用 `\{n,m\}` 可以精确匹配) `/lo\{1,3\}ng -> looong` (还匹配 `long`)
- ✦ (使用 `\{-n,m\}` 只匹配最少项) `/lo\{-1,3\} -> lo` (不匹配 `loo`)
- ✦ `[]` 匹配范围 `/[A-Z][a-z]\+ -> This` (不匹配 `is`)
- ✦ `^` 匹配行首 `/int main( -> int main(`
- ✦ `%` 特殊含义
- ✦ (`\%( \)` 和 `\( \)` 作用相同，但分组不编号)
- ✦ `/` 在匹配项上的偏移 `/main/e -> main` (光标移至单词结尾)
- ✦ `\?` 匹配零或一次 `/wo\?rd -> word` (还匹配 `wrd`)
- ✦ `~` 匹配最后一次的替换字符串
- ✦ `$` 匹配行尾 `/argv\[ \])$ -> argv[ ])`

向后搜索命令？

`?pattern` 从光标所在位置开始向前搜索与匹配模式 `pattern` 匹配的字符串。`vim` 找到匹配项时，光标移至匹配项的第一个字符。如果设置了选项 `hlsearch`，则匹配项高亮显示。

找到匹配项后，按下 `n` 键继续搜索下一个，按 `N` 反向搜索上一个。

```
89 /*
90
91 Function linear() is simple, it is used to calculate general linear result.
92
93 linear() conforms to the folloing formula:
94     y = f(x) = a + bx
95 there are three parameters in function linear() which named as a, b and x
96 corresponding to the above formula.
97
98 Among these three parameters, a is the constant term or the y intercept,
99 b is the coefficient of the independent variable, x is independent variable.
100 y or f(x) is dependent variable, and is returned as return value.
101
102 */
103 double linear(double a, double b, double x)
104 {
105     return a + b * x;
106 }
107
108 int main(int argc, char *argv[])
109     ?th\ (e\|ere\|ree\)
```

注：`\(` 和 `\)` 表示分组，`\|` 表示任取其一，因此上面的匹配模式会同时匹配单词 `the`、`there` 和 `three`。

r

当按下 `r` 键以后，下一个输入的字符会替换当前光标位置的字符。

如果 `r` 前面有重复因子 n ，则输入的字符会替换光标后连续 n 个字符。

`T h i s i s a v e r y v e r y l o n g w o r d .`

`rIwrAwrvwrVwrLwrW`



`T h i s I s A V e r y V e r y L o n g W o r d .`

R

当按下 R 键以后，后续输入的字符会替换当前光标位置以后的字符；到达行尾后如果继续输入，行会自动扩展。

如果 R 前面有重复因子 n ，则输入的字符在光标后连续替换 n 次。

This is a very very long word.

Rmuch longer word.<ESC>



This is a very much longer word.

替换命令 ..s

在命令模式下，`:s` 命令可用于文本替换，它的一般格式为：

`:m,n s/old/new/gc...`

这让 `vim` 在 `[m,n]` 行之间搜索字符串 `old`，并替换为 `new`。

- ✦ `m,n`
`m` 和 `n` 分别代表起始行（含）和结束行（含）。
(`n` 代表第 `n` 行，相当于 `n,n`)
(`.` 代表光标所在行)
(`%` 代表整个缓冲区)
(`$` 代表缓冲区末尾，因此 `1,$` 相当于 `%`)
- ✦ `s`
替换命令。
- ✦ `old`
原字符串。
- ✦ `new`
替换字符串。
- ✦ `gc...`
替换选项。
(`g` 如果没有该选项，只替换第一次出现的 `old` 字符串；否则，替换全部)
(`c` 在替换前提示用户确认)

替换命令示例 (1)

85 This is a very very long word.

:85s/very long/common/g



85 This is a very common word.

90 This is a long word.

91 This is a common word.

:90,91s/is/is NOT/g



90 This is NOT a long word.

91 This is NOT a common word.

93 This is a very long word.

:93s/^This/\/* This/

:93s/\.\$/. *\\//



93 /* This word is very long. */

替换命令示例 (2)

```
95 This is a very long word.
```

```
:95s/This \(.*\) a very long \(.*\)\/This \2 \1 very long./
```



```
95 This word is very long.
```

```
112 double linear(double a, double b, double x)
113 {
114     return a + b * x;
115 }
```

```
:112,115s/ \([abx]\)/ _\1/g
```



```
112 double linear(double _a, double _b, double _x)
113 {
114     return _a + _b * _x;
115 }
```