

# Vi 编辑器使用教程

原文链接: [文档](#)

Vi iMproved 6.3

## 一 在Vi 中移动光标

k	上
h	左
l	右
j	下
^	移动到该行第一个非空格的字符处
w	向前移动一个单词, 将符号或标点当作单词处理
W	向前移动一个单词, 不把符号或标点当作单词处理
b	向后移动一个单词, 把符号或标点当作单词处理
B	向后移动一个单词, 不把符号或标点当作单词处理
(	光标移至句首
)	光标移至句尾
{	光标移至段落开头
}	光标移至段落结尾
H	光标移至屏幕顶行
M	光标移至屏幕中间行
L	光标移至屏幕最后行
0	到行首
\$	到行尾
gg	到页首
G	到页末
行号+G	跳转到指定行
n+	光标下移n行
n-	光标上移n行
Ctrl+g	查询当前行信息和当前文件信息
fx	向右跳到本行字符x处 (x可以是任何字符)
Fx	向左跳到本行字符x处 (x可以是任何字符)
tx	和fx相同, 区别是跳到字符x前
Tx	和Fx相同, 区别是跳到字符x后
C-b	向上滚动一屏
C-f	向下滚动一屏
C-u	向上滚动半屏
C-d	向下滚动半屏
C-y	向上滚动一行
C-e	向下滚动一行
nz	将第n行滚至屏幕顶部, 不指定n时将当前行滚至屏幕顶部。

## 二 进入和退出Vi命令

<code>vi filename</code>	打开或新建文件，并将光标置于第一行首
<code>vi +n filename</code>	打开文件，并将光标置于第n行首
<code>vi + filename</code>	打开文件，并将光标置于最后一行首
<code>vi +/pattern filename</code>	打开文件，并将光标置于第一个与pattern匹配的串处
<code>vi -r filename</code>	在上次正用vi编辑时发生系统崩溃，恢复filename
<code>vi filename ... filename</code>	打开多个文件，依次进行编辑
<code>ZZ</code>	退出vi并保存
<code>:q!</code>	退出vi，不保存
<code>:wq</code>	退出vi并保存

## 三 重复操作

<code>.</code>	重复上一次操作
----------------	---------

## 四 自动补齐

<code>C-n</code>	匹配下一个关键字
<code>C-p</code>	匹配上一个关键字

## 五 插入

<code>o</code>	在光标下方新开一行并将光标置于新行行首，进入插入模式。
<code>O</code>	同上，在光标上方。
<code>a</code>	在光标之后进入插入模式。
<code>A</code>	同上，在光标之前。
<code>R</code>	进入替换模式，直到按下Esc
<code>set xxx</code>	设置XXX选项。

## 六 行合并

<code>J</code>	把下面一行合并到本行后面
----------------	--------------

## 七 Vi 中查找及替换命令

/pattern	从光标开始处向文件尾搜索pattern
?pattern	从光标开始处向文件首搜索pattern
n	在同一方向重复上一次搜索命令
N	在反方向上重复上一次搜索命令
%	查找配对的括号
:s/p1/p2/g	将当前行中所有p1均用p2替代，若要每个替换都向用户询问则应该用gc选项
:n1,n2s/p1/p2/g	将第n1至n2行中所有p1均用p2替代
:g/p1/s//p2/g	将文件中所有p1均用p2替换

.\*[]^%~\$ 在vi中具有特殊含义，若需要查找则应该加上转义字符"\\"

## 1 查找的一些选项

### 设置高亮

:set hlsearch	设置高亮
:set nohlsearch	关闭高亮
:nohlsearch	关闭当前已经设置的高亮

### 增量查找

:set incsearch	设置增量查找
:set noincsearch	关闭增量查找

## 八 在Vi中删除

x	删除当前光标下的字符
dw	删除光标之后的单词剩余部分。
d\$	删除光标之后的该行剩余部分。
dd	删除当前行。
c	功能和d相同，区别在于完成删除操作后进入INSERT MODE
cc	也是删除当前行，然后进入INSERT MODE

## 九 更改字符

rx	将当前光标下的字符更改为x（x为任意字符）
~	更改当前光标下的字符的大小写

## 十 键盘宏操作

qcharacter	开始录制宏，character为a到z的任意字符
q	终止录制宏
@character	调用先前录制的宏

## 十一 恢复误操作

u	撤销最后执行的命令
U	修正之前对该行的操作
Ctrl+R	Redo

## 十二 在 Vi 中操作 Frame

c-w c-n	增加frame
c-w c-c	减少frame
c-w c-w	切换frame
c-w c-r	交换两个frame

## 十三 VIM 中的块操作

Vim支持多达26个剪贴板。

选块	先用v, C-v, V选择一块, 然后用y复制, 再用p粘贴。
yy	复制当前整行
nyy	复制当前行开始的n行内容
?nyy	将光标当前行及其下n行的内容保存到寄存器?中, 其中?为一个字母, n为一个数字
?nyw	将光标当前行及其下n个词保存到寄存器?中, 其中?为一个字母, n为一个数字
?nyl	将光标当前行及其下n个字符保存到寄存器?中, 其中?为一个字母, n为一个数字
?p	将寄存器?中的内容粘贴到光标位置之后。如果?是用yy复制的完整行, 则粘贴在光标所在行下面。这里?可以是一个字母, 也可以是一个数字
?P	将寄存器a中的内容粘贴到光标位置之前。如果?是用yy复制的完整行, 则粘贴在光标所在行上面。这里?可以是一个字母, 也可以是一个数字
ay[motion]	
ay\$	复制光标位置到行末并保存在寄存器a中
ayft	复制光标位置到当前行第一个字母t并保存在寄存器a中

以上指令皆可去掉 a 工作, 则 y,p 对未命名寄存器工作 (所有 d,c,x,y 的对象都被保存在这里)。

### 1 剪切/复制/粘贴

所有删除的内容自动被保存, 可以用 p 键粘贴。

## 十四 Vi 的选项设置

all	列出所有选项设置情况
term	设置终端类型
ignorance	在搜索中忽略大小写
list	显示制表位(Ctrl+I)和行尾标志(\$)
number	显示行号
report	显示由面向行的命令修改过的数目