

基本操作

移动光标

l 右移一个字符；
 h 左移一个字符；
 k 上移一行；
 j 下移一行；
 w 右移一个单词，光标在词首¹；
 b 左移一个单词，光标在词首¹；
) 右移一句，光标在句首；
 (左移一句，光标在句首；
 { 上移一段，光标在空白行；
 } 下移一段，光标在空白行；
 fx 在当前行向右搜索 x 字符，光标在 x 字符上²；
 Fx 在当前行向左搜索 x 字符，光标在 x 字符上²；
 ^ 移动到当前行第一个字符，除空格外；
 O 移动到当前行行首；
 \$ 移动到当前行行尾；
 gk 上移一行，包括自动换行³；
 gj 下移一行，包括自动换行³；
 [n]G 移动到第 n 行⁴；
 :[n]<cr> 移动到第 n 行⁵；
 gg 移动到第一行；
 G 移动到最后一行；
 % 在匹配的字符间跳转⁶；
 H 移动到屏幕显示的第一行；
 M 移动到屏幕显示的中间一行；
 L 移动到屏幕显示的最后一行；

插入文本

a 在光标右边插入文本⁷；
 A 在当前行行尾插入文本；
 i 在光标左边插入文本⁷；
 I 在当前行行首插入文本；
 O 在当前行上边新建一行插入；
 o 在当前行下边新建一行插入；
 :r filename 在当前位置插入另一个文件的内容；
 :[n]r filename 在第 n 行下插入另一个文件的内容⁴；
 :r !command 在当前行下边新建一行并插入 shell 命令的输出信息；

替换

r 替换光标处的字符⁸；
 R 进入替换模式⁹；
 :s/old/new 用 new 替换当前行第一个 old¹⁰；
 :%s/^/xxx/g 在每一行的行首插入 xxx¹¹；

复制与粘贴

x 剪切光标处的字符¹²；
 [n]x 剪切右边 n 个字符；
 [n]X 剪切左边 n 个字符；
 y 复制选中的内容¹³；
 yy or Y 复制整行内容¹⁴；
 d 删除选中的内容^{12,13}；
 dd 删除整行内容¹⁴；
 d[n]w 删除 n 个单词^{*15}；
 :[n],[m]d<cr> 删除 n 行到 m 行的内容¹⁵；

d1G or dgg 删除当前行及上边所有行¹⁵；
 dG 删除当前行及下边所有行¹⁵；
 d/x<cr> 删除光标位置到下一个 x 字符之间的内容^{2,15}；
 P 在光标之左粘贴；
 p 在光标之右粘贴；
 c 修改光标处字符¹⁶；
 cc 修改当前行；
 c[n]w 修改光标后 n 个单词^{*4}；

撤销与重做

u 撤销上一次改动；
 Ctrl+r 重做上一次改动¹⁷；
 U 撤销当前行中所有改动；
 [n]u 撤销上 n 次改动⁴；
 :earlier 4m 回到四分钟前；
 :later 55s 前进五十五秒；
 :undolist 撤销历史的记录；

查找

/something 在右边的文本中查找 something；
 ?something 在左边的文本中查找 something；
 n 向后查找下一个；
 N 向前查找下一个；

翻页

Ctrl+b 上翻一屏¹⁷；
 Ctrl+f 下翻一屏¹⁷；
 Ctrl+u 上翻半屏¹⁷；
 Ctrl+d 下翻半屏¹⁷；

Ctrl+y 上翻一行¹⁷；
 Ctrl+e 下翻一行¹⁷；
 zz: 将当前行移动到屏幕中央；
 zt: 将当前行移动到屏幕顶端；
 zb: 将当前行移动到屏幕底端；

基本排版

>> 在行首向右缩进；
 << 在行首向左缩进；
 gq 对选中的文字重排¹⁸；
 gq[n]j 重排下面 n 行⁴；
 gqG 重排当前行及下边所有行；
 J 拼接当前行和下一行；

文档操作

:e newfile 关闭当前编辑的文件，并打开 newfile¹⁹；
 :e + newfile 从文件的末尾开始打开 newfile；
 :e +[n] newfile 从第 n 行开始打开 newfile⁴；
 :e 重新加载当前文件¹⁹；
 :e# or ctrl+^ 回到上一个编辑的文件；
 :f or ctrl+g 显示文档名和光标位置等信息；
 :f filename 改变编辑的文件名，但文件并没有存储；
 :w 保存修改；
 :[n],[m]w filename 保存 n 行到 m 行的内容为新文件⁷；
 :q 退出当前窗口；
 :wq or :x 保存并退出；
 ZZ 保存并退出；

:enew 新建一个未命名的文件；
 :saveas newfile 另存为；

启动

vim filename 打开文件；
 vim -r filename 恢复上次异常退出的文件；
 vim -R filename 以只读的方式打开文件，但可以强制保存²⁰；
 vim + filename 从文件的末尾开始打开 filename；
 vim +[n] filename 从第 n 行开始打开 filename⁷；
 vim +/string filename 打开 filename，并将光标停留在第一个找到的 string 上；

模式之解

Normal mode (常规模式) 对光标进行移动、复制、粘贴等命令。无论在任何模式下，按 <Esc> 键都可以回到常规模式；
Insert mode (插入模式) 输入和编辑文件内容。在常规模式下，按 a、A、i、I、o、O、c 等键进入插入模式；

Last line mode (命令行模式) 对文件进行复制、插入、搜索、替换和更多的高级操作。在常规模式下，按 :、/、? 键进入命令行模式；

Visual mode (可视模式) 对文件进行可视化的选取，然后在进行操作。在常规模式下，按 v、Ctrl+v 等键进入可视模式；

Replace mode (替换模式) 替换文本内容。在常规模式下，按 r、R 键进入可视模式；

*译注：h、j、k、l、w、b、\$.0 等命令同样适用；
 1译注：还有两个命令与此类似，e 右移一个单词，光标在词尾；ge 左移一个单词，光标在词尾；
 2译注：x表示任意字符，支持汉字。还有两个命令与此类似，tx 在当前行向右搜索 x，光标在 x 字符前；Tx 在当前行向左搜索 x，光标在 x 字符前；
 3译注：gj、gk命令以屏幕显示为准，即包括文件自动换行的情况，还有两个命令与此类似，g0 移动到所在屏幕行行首；g\$ 移动到所在屏幕行行尾；
 4译注：[n]、[m] 为任意阿拉伯数字；
 5译注：前边加冒号进入命令行模式，<cr> 表示键盘上的 Enter 键；:\$<cr> 移动到最后一行；
 6译注：匹配符号包括()、{}、[]、<>等；
 7译注：小技巧，按 3，再按 i 或 a，进入插入模式，输入 =，然后按 Esc 键进入常规模式，就会出现 ===。这在插入分割线时非常有用，如 36i + <Esc>插入 36 个 + 组成的分割线；
 8译注：支持汉字；
 9译注：按 <Esc> 键回到常规模式；
 10译注：:s/old/new/g 用 new 替换文件中所有的 old，% 表示整个文档；
 11译注：^ 表示行首，所有替换命令末尾加上 c，表示每个替换都需要用户确认。如：%s/old/new/gc，加 i 则表示忽略大小写；
 12译注：vim 中剪切和删除是一个概念，都是把当前内容存储到寄存器中；
 13译注：按 v 键进入可视模式。移动光标以覆盖想编辑的内容(被选中的内容会以高亮显示)，最后键入操作符命令。用 Ctrl+v 进入可视模式，以矩形文本块为对象进行选择(这在编辑表格时会派上大用场)，光标可以在矩形的四个角之间跳转，o 光标在矩形的对角之间跳转；O 光标在矩形的左右角之间跳转；
 14译注：[n]yy 复制 n 行内容；[n]dd 删除 n 行内容；
 15译注：把 d 命令换成 y 命令同样适用；
 16译注：先删除光标处字符，然后进入插入模式；
 17译注：Ctrl 为键盘上的 Control 键；
 18译注：对过长的字符串进行断行，对过短的字符串进行合并行。
 19译注：如对当前文件的修改未保存，vim 会警告，而且不会对文件执行退出、打开等命令，在命令行模式中加入 ! 表示强制命令，在命令不执行的情况下可以加入 ! 强制执行命令，但后果要自负；:e! newfile 强制退出当前文件，并打开 newfile
 20译注：vim -M filename 以只读的方式打开文件，不可以强制保存；