

左右移动

状态	命令	描述
✔	 h	左移 (同功能: CTRL-H, BS, or Left key)
✔	 l	右移 (同功能: Space or Right key)
✔	0	移动到当前行的第一个字符处 (同功能: Home key)
✔	^	移动到当前行的第一个非空字符处
✔	 s	移动到当前行的最后一个字符处 (N-1 lines lower) (同功能: End key)
✔	gm	移动到屏幕上显示行的中央
✔	 \	移动到指定列 (默认: 1)
✔	 f{char}	向右移动到第 N 个指定字符处
✔	 F{char}	向左移动到第 N 个指定字符处
✔	 t{char}	向右移动到第 N 个指定字符的前一个字符处
✔	 T{char}	向左移动到第 N 个指定字符的前一个字符处
✔	 ;	重复执行 N 次上一次的 “f”, “F”, “t”, 或 “T” 命令
✔	 ,	反向重复执行 N 次上一次的 “f”, “F”, “t”, 或 “T” 命令

上下移动

状态	命令	描述
✔	 k	上移 (同功能: CTRL-P and Up)
✔	 j	下移 (同功能: CTRL-J, CTRL-N, NL, and Down)
✔	 -	上移, 光标将位于第一个非空白字符上
✔	 +	下移, 光标将位于第一个非空白字符上 (同功能: CTRL-M and CR)
✔	 _	下移 N-1 行, 光标将位于第一个非空白字符上
✔	 G	移动到第 N 行, 光标将位于第一个非空白字符上(默认: 最后一行)
✔	 gg	移动到第 N 行, 光标将位于第一个非空白字符上(默认: 第一行)
✔	 %	移动到当前文件的第 N%行, 必须指定 N, 否则将执行“%”命令
✔	 gk	上移 N 行(当有折叠行时与‘k’命令的行为不同, 折叠行被视作一行)
✔	 gi	下移 N 行(当有折叠行时与‘j’命令的行为不同, 折叠行被视作一行)

针对文本对象的移动方式

状态	命令	描述
✔	 w	向前移动 N 个单词
✔	 W	向前移动 N 个单词, 忽略分隔符
✔	 e	向前移动 N 个单词, 光标位于第 N 个单词的结尾
✔	 E	向前移动 N 个单词, 光标位于第 N 个单词的结尾, 忽略分隔符
✔	 b	向后移动 N 个单词
✔	 B	向后移动 N 个单词, 忽略分隔符
✔	 ge	向后移动 N 个单词, 光标位于第 N 个单词的结尾
✔	 gE	向后移动 N 个单词, 光标位于第 N 个单词的结尾, 忽略分隔符
✔	)	向前移动 N 个句子
✔	 (向后移动 N 个句子
✔	 }	向前移动 N 个段落
✔	 {	向后移动 N 个段落
✔	]]	向前移动 N 个缓冲区,光标位于开始位置
✔	 [[向后移动 N 个缓冲区,光标位于开始位置
✔	]]	向前移动 N 个缓冲区,光标位于结束位置
✔	 []	向后移动 N 个缓冲区,光标位于结束位置
✔	 [(向后移动到第 N 个未闭合的‘(’处
✔	 [(向后移动到第 N 个未闭合的‘(’处
⬇	 [m	向后移动到第 N 个方法的开始位置(Java)
⬇	 [M	向后移动到第 N 个方法的结束位置(Java)
✔	)]	向前移动到第 N 个未闭合的‘)’处
✔	)]	向前移动到第 N 个未闭合的‘)’处
⬇	]m	向前移动到第 N 个方法的开始位置(Java)
⬇	]M	向前移动到第 N 个方法的结束位置(Java)
⬇	 [#	向后移动到第 N 个未匹配的 #if、#else
⬇	 #	向前移动到第 N 个未匹配的 #else、#endif
⬇	 [*	向后移动到第 N 个 C 注释的开始位置
⬇	 *	向前移动到第 N 个 C 注释的开始位置



按模式搜索

状态	命令	描述	备注
✔🌟	 /{pattern}[/[offset]]<CR>	向前搜索{pattern}的第 N 次出现	当前仅支持 JavaScript 的正则引擎, 不支持 Vim 的内置正则引擎
✔🌟	 ?{pattern}[?[offset]]<CR>	向后搜索{pattern}的第 N 次出现	当前仅支持 JavaScript 的正则引擎, 不支持 Vim 的内置正则引擎
✔	 n	重复上一次搜索	
✔	 N	反方向重复上一次搜索	
✔	 *	向前搜索当前光标所处的单词	
✔	 #	向后搜索当前光标所处的单词	
✔	 g*	类似于 “*”, 执行部分匹配	
✔	 g#	类似于 “#”, 执行部分匹配	
✔	gd	跳转到当前光标所处标识的声明处	
⬇	gD	跳转到当前光标所处标识的声明处	

标记定位

状态	命令	描述
✔	m{a-zA-Z}	使用{a-zA-Z}标记当前位置
✔	`{a-z}	跳转到当文件中的{a-z}标记处
✔	`{A-Z}	跳转到任意文件中的(A-Z)
✔	`{0-9}	跳转到 Vim 上次退出时的位置
✔	``	跳转到 Vim 最后一次跳转之前的位置
⬇	``	跳转到当前文件中最后一次编辑的位置
⬇	`[跳转到上一次操作或输入文本的开始位置
⬇	`]	跳转到上一次操作或输入文本的结束位置
✔	`,`	跳转到此文件的最后一次修改处
✔	`,`	跳转到此文件的最后一次修改处





其它移动方式

状态	命令	描述
✔	%	在当前行中查找下一个大括号, 中括号, 小括号或者“#if/ “#else/”“#endif”,并且跳转到与之匹配的结束标记处
✔	 H	跳转到距离视口首行第 N 行的第一个非空字符处
✔	M	跳转到视口中央行的第一个非空字符处
✔	 L	跳转到距离视口最后一行第 N 行的第一个非空字符处

滚动

状态	命令	描述
✔	 CTRL-E	向下滚动 N 行(默认: 1)
✔	 CTRL-D	向下滚动 N 个 1/2 屏(默认: 1/2 屏)
✔	 CTRL-F	向下滚动 N 屏
✔	 CTRL-Y	向上滚动 N 行(默认: 1)
✔	 CTRL-U	向上滚动 N 个 1/2 屏(默认: 1/2 屏)
✔	 CTRL-B	向上滚动 N 屏
✔	zt	将当前行移到屏幕顶部
✔	zz	将当前行移到屏幕中央
✔	zb	将当前行移到屏幕底部

以下命令仅在换行关闭时有效:

状态	命令	描述	备注
✔🌟	 zh	向右滚动 N 个字符	在 VSCode 中, 当运行此命令时, 无论水平滚动条是否移动, 光标总会移动
✔🌟	 zl	向右滚动 N 个字符	同上
✔🌟	 zH	向右滚动半个屏幕宽度	同上
✔🌟	 zL	向左滚动半个屏幕宽度	同上





插入文本

状态	命令	描述
✔	 a	在光标后方插入文本
✔	 A	在行尾插入文本
✔	 i	在光标前方插入文本
✔	 I	在本行第一个非空字符前插入文本
✔	 gl	在第一列前插入文本
✔	gi	在最后一次改动处插入文本
✔	 o	在当前行的下方插入新行
✔	 O	在当前行的上方插入新行

删除文本

状态	命令	描述
✔	 x	从光标所处的当前字符开始删除 N 个字符
✔	 X	删除光标前的 N 个字符
✔	 dd	删除 N 行
✔	 D	删除到行尾
✔	 J	连接当前行和下一行(删除空行)
✔	 gJ	和“J”命令相同, 但是不会在连接处插入空格

文本复制和移动

状态	命令	描述
✔	:reg	显示寄存器中的所有内容
✔	:reg {arg}	显示寄存器中{arg}的内容
✔	 yy	把当前行开始的 N 行放入寄存器(包含当前行)
✔	 Y	把当前行开始的 N 行放入寄存器(包含当前行)
✔	 p	把寄存器中的内容放置到光标后方(执行 N 次)
✔	 P	把寄存器中的内容放置到光标前方(执行 N 次)


修改文本

状态	命令	描述	备注
✔	 r{char}	使用{char}替换 N 个字符	
✔🌟	 R	进入替换模式(重复 N 次输入的文本)	不支持{count}
✔🌟	 R	进入替换模式(重复 N 次输入的文本)	不支持{count}

(以下命令为删除文本同时进入插入模式)

状态	命令	描述
✔	 c{motion}	修改移动命令{motion}经过的文本
✔	 cc	修改 N 行(包含当前行)
✔	 S	修改 N 行(包含当前行)
✔	 C	修到行尾的内容(包括 N-1 行)
✔	 s	修改 N 个字符
✔	 <<	将 N 行向左缩进
✔	 >>	将 N 行向右缩进

重复性命令

状态	命令	描述	备注
✔🌟	 .	重复最后一次修改(N:重复次数)	未发生在光标上的修改无法重复
✔	q{a-z}	重复寄存器中{a-z}的字符	
⬇	q{A-Z}	记录输入的字符,放入寄存器中,对应的标记为小写的{a-z}	
✔	q	停止记录	

撤消/恢复 命令

状态	命令	描述	备注
✔	 u	撤消前 N 次修改	目前的实现可能无法完全涵盖所有情况
✔	 CTRL-R	恢复前 N 次撤消的修改	同上
✔	U	恢复上一次修改过的行	

编辑文件

状态	命令	描述	备注
✔🌟	:e[dit] {file}	编辑 {file}.	将在当前分组编辑器的新选项卡中打开文件, 而不是在当前选项卡中打开

折叠

状态	命令	描述
⬇	zf{motion} or {Visual}zf	创建折叠
⬇	zF	折叠[count]行. 类似于“zf”.
⬇	zd	删除当前光标下的折叠内容
⬇	zD	递归删除当前光标下所有的折叠内容
⬇	zE	打开窗口中所有的折叠内容
✔	zo	打开光标下的折叠内容,当指定数量时,将打开多个折叠内容
✔	zO	递归打开当前光标下所有的折叠内容
✔	zc	在光标下关闭一个折叠,当给出计数时, 关闭多个折叠
✔	zC	递归关闭当前光标下所有的折叠内容
⬇	za	处于关闭的折叠块时,打开折叠块.反之,关闭折叠块
⬇	zA	处于关闭的折叠块时,递归的打开折叠块.反之,递归的关闭折叠块
⬇	zv	查看光标所在行: 打开刚好足够的折叠, 使光标所在行不折叠
⬇	zx	更新折叠: 撤消手动打开和关闭折叠: 重新应用'foldlevel', 然后执行“zv”: 查看光标行
⬇	zX	撤消手动打开和关闭折叠
⬇	zm	折叠更多: 从'foldlevel'中减去一个
✔	zM	关闭所有的折叠: 将'foldlevel'设为 0. 将设置'foldenable'
⬇	zr	减少折叠: 在'foldlevel'中添加一个
✔	zR	打开所有的折叠. 会将'foldlevel'调整到最高级.
⬇	zn	不折叠: 重置'foldenable'. 所有的折叠都会打开
⬇	zN	正常折叠: 设置'foldenable'.所有折叠都将像以前一样.
⬇	zi	反转'foldenable'
⬇	[z	移动到当前打开的折叠块的首部
⬇]z	移动到当前打开的折叠块的尾部
⬇	zj	向下移动到下一个折叠的开始
⬇	zk	向上移动到上一个折叠的开始