

Vim Cheat Sheet

Navigation

h - 左移光标 **j** - 下移光标 **k** - 上移光标 **l** - 右移光标

w - 移动到下个单词开头 **W** - 移动到下个单词开头(单词含标点)

e - 移动到下个单词结尾 **E** - 移动到下个单词结尾(单词含标点)

b - 移动到上个单词开头 **B** - 移动到上个单词开头(单词含标点)

% - 跳转到配对的符号(默认支持的配对符号组: '()', '{}', '[]')

o - 移动到行首 **^** - 移动到行首的非空白符

\$ - 移动到行尾 **g_** - 移动到行内最后一个非空白符

gg - 移动到文件第一行 **G** - 移动到文件最后一行

5gg or **5G** - 移动到第五行

gd - move to local declaration

gD - move to global declaration

fx - 移动到字符 x 下次出现的位置

Fx - 移动到字符 x 上次出现的位置

tx - 移动到字符 x 下次出现的位置的前一个字符

Tx - 移动到字符 x 上次出现的位置的后一个字符

; - 重复之前的 **f**、**t**、**F**、**T** 操作

, - 反向重复之前的 **f**、**t**、**F**、**T** 操作

} - 移动到下一个段落 (当编辑代码时则为函数 / 代码块)

{ - 移动到上一个段落 (当编辑代码时则为函数 / 代码块)

zz - 移动屏幕使光标居中

Ctrl + e - 向下移动屏幕一行(保持光标不动)

Ctrl + y - 向上移动屏幕一行(保持光标不动)

Ctrl + b - 向上滚动一屏

Ctrl + f - 向下滚动一屏

Ctrl + d - 向下滚动半屏

Ctrl + u - 向上滚动半屏

Tip 命令前追加数字表示命令的重复次数, 比如 **4j** 表示向下移动四行.

Insertion

i - 从光标前开始插入字符 **I** - 从行首开始插入字符

a - 从光标后开始插入字符 **A** - 从行尾开始插入字符

o - 在当前行之下另起一行, 开始插入字符

O - 在当前行之上另起一行, 开始插入字符

ea - 从当前单词末尾开始插入

Ctrl + h - 在插入模式下 · 删除光标前的字符

Ctrl + w - 在插入模式下 · 删除光标前的单词

Ctrl + j - 在插入模式下 · 另起一行

Ctrl + t - 在插入模式下 · 向右缩进 · 宽度由 **shiftwidth** 控制

Ctrl + d - 在插入模式下 · 向左缩进 · 宽度由 **shiftwidth** 控制

Ctrl + n - 在插入模式下 · 在光标之前插入自动补全的下一个匹配项

Ctrl + p - 在插入模式下 · 在光标之前插入自动补全的上一个匹配项

Ctrl + rx - insert the contents of register x

Esc - 退出插入模式

Editing

r - 替换当前字符

J - 将下一行合并到当前行, 并在两部分文本之间插入一个空格

gJ - 将下一行合并到当前行, 两部分文本之间不含空格

gwip - 重新调整段落

cc - 将光标所在的行删除, 然后进入插入模式

C - 将光标处到行尾删除, 然后进入插入模式

c\$ - 将光标处到行尾删除, 然后进入插入模式

ciw - 将光标所在的单词删除, 然后进入插入模式

cw - 从光标位置开始, 修改单词

s - 删除当前字符, 然后进入插入模式

S - 清空当前行, 然后进入插入模式 (同 **<kbd>cc</kbd>**)

xp - 当前字符后移

u - 撤销 **U** - restore (undo) last changed line

Ctrl + r - 重复 **.** - 再次执行上个命令

g~ - switch case up to motion
gu - change to lowercase up to motion
gU - change to uppercase up to motion

Visual Commands (Visual Mode)

v - 进入可视化模式, 移动光标高亮选择, 然后可以对选择的文本执行命令(
V - 进入可视化模式(行粒度选择)
Ctrl + v - 进入可视化模式(矩阵选择)
o - 切换光标到选择区开头/结尾
O - 切换光标到选择区的角
aw - 选择当前单词
ab - 选择被 () 包裹的区域(含括号)
aB - 选择被 {} 包裹的区域(含花括号)
at - 选择被 <> 标签包裹的区域(含<>标签)
ib - 选择被 () 包裹的区域(不含括号)
iB - 选择被 {} 包裹的区域(不含花括号)
it - 选择被 <> 标签包裹的区域(不含<>标签)
Esc - 退出可视化模式

> - 向右缩进	< - 向左缩进
y - 复制	d - 剪切
~ - 大小写切换	
u - 将选中文本转换为小写	U - 将选中文本转换为大写

Tip Instead of b or B one can also use (or { respectively.

Cut and paste

yy - 复制当前行	2yy - 复制 2 行
yw - 复制当前单词	y\$ - 复制, 从光标位置到行末
p - 在光标后粘贴	P - 在光标前粘贴
dd - 剪切当前行	2dd - 剪切 2 行
dw - 剪切当前单词	D - 剪切, 从光标位置到行末
d\$ - 剪切, 从光标位置到行末 (同<kbd>D</kbd>)	
x - 剪切当前字符	

Search and replace

/pattern - 查找<kbd>pattern</kbd>
?pattern - 向上查找<kbd>pattern</kbd>
\vpattern - <kbd>pattern</kbd> 中的非字母数字字符被视为正则表达式特殊字符
n - 查找下一个 N - 查找上一个
:s/old/new/g - 替换全部 :s/old/new/gc - (逐个)替换
:noh[search] - 移除搜索结果的高亮显示

Search in multiple files

:vim[grep] /pattern/ {file} - 在多个文件中搜索 <kbd>pattern</kbd>
e.g. :vim[grep] /foo/ **/*
:cn[ext] - 移动至下一个 :cp[revious] - 移动至上一个
:cope[n] - 打开搜索结果列表
:ccl[ose] - close the quickfix window

Indent text

>> - 将当前行向右缩进 · 宽度由 shiftwidth 控制
<< - 将当前行向左缩进 · 宽度由 shiftwidth 控制
>% - 向右缩进 () 或 {} 内的区域 (光标需置于括号上)
>ib - 向右缩进 () 内的区域
>at - 向右缩进 <> 标签内的区域
3== - 自动缩进 3 行
=% - 自动缩进 () 或 {} 内的区域 (光标需置于括号上)
=iB - 自动缩进 {} 内的区域 (光标需置于括号上)
gg=G - 自动缩进整个缓冲区
]p - 粘贴并调整缩进至当前行

Macros

qa - 录制宏 a
q - 停止录制宏
@a - 执行宏 a
@@ - 重新执行上次执行的宏

Registers

:reg[isters] - 显示寄存器内容

"xy - 复制内容到寄存器 x

"xp - 粘贴寄存器 x 中的内容

"+y - 复制内容到系统剪贴板寄存器

"+p - 粘贴系统剪贴板寄存器的内容

Tip 寄存器被存储在 ~/.viminfo 中, 在下次重启 vim 时仍会加载

Tip 特殊寄存器 :

0 - 上次复制

" - 未命名寄存器 · 上次复制或删除

% - 当前文件名

- 轮换文件名

* - 剪贴板内容 (X11 primary)

+ - 剪贴板内容 (X11 clipboard)

/ - 上次搜索的 pattern

: - 上次执行的命令

. - 上次插入的文本

-- 上次剪切的短于一行的文本

= - 表达式寄存器

_ - 黑洞寄存器

Marks

:marks - 标记列表

ma - 设置当前位置为标记 a

`a - 跳转到标记 a 的位置

y`a - 复制当前位置到标记 a 的内容

`0 - go to the position where Vim was previously exited

`` - go to the position when last editing this file

`. - go to the position of the last change in this file

`` - go to the position before the last jump

:ju[mps] - list of jumps

Ctrl + i - go to newer position in jump list

Ctrl + o - go to older position in jump list

:changes - list of changes

g, - go to newer position in change list

g; - go to older position in change list

Ctrl +] - jump to the tag under cursor

Tip To jump to a mark you can either use a backtick (`) or an apostrophe ('). Using an apostrophe jumps to the beginning (first non-black) of the line holding the mark.

Tabs

:tabnew or :tabnew {page.words.file} - 在新标签中打开文件

Ctrl + wT - 将窗口变成标签

gt or :tabn[ext] - 切换到下一个标签

gT or :tabp[revious] - 切换到上一个标签

#gt - 切换到第 <kbd>#</kbd> 个标签

:tabm[ove] # - 移动标签到第 <kbd>#</kbd> 位(下标从 0 开始)

:tabc[lose] - 关闭当前标签

:tabo[nly] - 关闭其他标签

:tabdo command - 在所有标签中执行命令 (例如 :tabdo q 关闭所有标签)

Exit

:w - 保存

:w !sudo tee % - 使用 sudo 保存当前文件

:wq or :x or ZZ - 保存并退出

:q - 退出(修改未保存时警告)

:q! or ZQ - 不保存强制退出

:wqa - 保存所有标签页并全部退出