



链滴

VIM 实战笔记

作者: [ZephyrJung](#)

原文链接: <https://ld246.com/article/1521526619028>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

记录了《VIM实战》中的常用指令，高级主题部分未涉及

模式

普通模式

. 重复上次修改

u 撤销

dd 删除整行

>G 当前行到文档末尾向右缩进, >> 当前行向右缩进。反之用<

A 移动到行尾并插入 <=> \$a

/pattern<CR> 文档中查找下一处匹配项

?pattern<CR> 文档中查找上一处匹配项

s 删除当前位置的字符并进入插入模式

:s/target/replacement 将当前行的target替换为replacement, 通过&重复执行

* 定位到下一个当前单词通过:set hls 高亮, :set hls!取消高亮 (其他单词匹配指令类似)

o 在当前行的下一行插入新的一行并进入插入模式

I 定位到行首并进入插入模式

daw delete a word, 删除光标所在单词。

<C-a> 当前数字加1, <C-x> 当前数字减1, 10<C-a>当前数字加10

yyp 复制当前行

gUaw 当前单词大写, guaw 当前单词小写。gUU 当前行大写, guu 当前行小写, g~ 反转大小写

插入模式

<C-h> 删除前一个字符, <C-w> 删除前一个单词, <C-u> 删至行首

yt, 复制当前到逗号之前的文本到寄存器中, 插入模式下通过 <C-r>0 粘贴

<C-r>=6*35<CR> 在当前位置插入计算结果, <CR> 代表回车

<C-k>12 插入二分之一, <C-k>?! 输入倒过来的问好 (二合字母, 可通过:digraphs 或:h digraph-true查看)

R 插入模式进入替换模式, 相当于INSERT键

可视模式

v 激活面向字符的可视模式, V 面向行, <C-v> 面向块, gv 重选上次高亮选区, o 在选区两端切换

Vr- 用 - 替换当前行

列编辑实例: <C-v>jje 选中多列单词, c 删除单词, 并进入插入模式, 键入新单词后, 返回普通模式即可。(编辑时, 只会看到第一行上看到效果); <C-v>jj\$A; 在多列行尾添加分号

命令行模式

:**[range]delete[x]** 删除指定范围内的行[到寄存器x中]

:**[range]yank[x]** 复制指定范围的行[到寄存器x中]

:**[line]put[x]** 在指定行后粘贴寄存器x中的内容

:**[range]copy{address}** 把指定范围内的行复制到{address}所指定的行之下

:**[range]move{address}** 把指定范围内的行移动到{address}所指定的行之下

:**[range]join** 连接指定范围内的行

:**[range]normal{commands}** 对指定范围内的每一行智行普通模式命令{commands}

:**[range]substitute/{pattern}/{string}/[flags]** 把指定范围内出现{pattern}的地方替换为{string}

:**[range]global/{pattern}/[cmd]** 对指定范围内匹配{pattern}的所有行, 在其上执行Ex命令{cmd}

<C-r> {register} 把任意寄存器的内容插入到命令行

:1 跳转到首行, :\$ 跳转到文件末尾

:p 或 :print 在屏幕上打印当前行, :1p 打印行1, :2,5p 打印2~5行, :.,\$p 打印当前行到末行, :%p 印所有行

:s 或 :substitute 替换, :%s/Practical/Pragmatic/ 将每一行的第一个Practical替换为Pragmatic

可以通过高亮选区指定命令范围, 如先执行2G, 在执行VG, 即可选中从第2行到文本尾处, 再按下: 会预先填充一个范围

可以通过模式指定范围, 如: /<html> /, /</html> /p, 可以通过偏移排除上下两行, 如: /<html> /+1, /</html> /-1p, 同理可以对行号偏移, 假定当前行为2, :.,+3 相当于 :2,5

:copy 或 :co 或 :t 复制, 如 :6copy. 将第6行拷贝至当前行, t6 将当前行复制到第6行下方, t. 为当前创建一个副本, 类似于ypp, t0 复制当前行到开头

:move 或 :m 移动行

@: 重复执行上一次的命令

:normal 对范围内的所有行同时执行某命令, 如jVG 选中行, :<,'>normal. 对高亮的每一行都执行命令。:%normal A; 对全文的每一行后面加分号。

输入指令时, 可通过<C-d> 显示补全选项, 如: colorscheme <C-d> 可以看到当前可选的配色方案。ab键可以补全正确的指令, 如: edit 或 :write 将补全当前目录中的目录或文件名, :tab 补全标签名, :s t 补全设置。

`<C-r> <C-w>` 复制光标下的单词到命令行中

命令模式下，可以通过上下键浏览历史命令

备注

- 所有进入插入模式的命令
- 所有定位命令

文件

管理多个文件

`vim *.txt` 打开多个文件。`:ls` 显示当前文件列表缓冲区。`:bnext` 跳转到下一个缓冲区，`:bprev` 跳转到一个缓冲区。`:bfirst` 跳转到开头，`:blast` 跳转到结尾。通过 `<C-^>` 在标记为%的当前文件和# 文件切换。`:buffer N` 直接跳转到编号为N的缓冲区。`:buffer {bufname}` 跳转到指定名称的缓冲区，可以通过**T** b补全。`:bufdo` 对缓冲区列表中的所有文件执行命令。

已打开vim编辑器时，可以通过`:args index.html app.js` 来想缓冲区中添加文件

打开及保存文件

`:edit {filename}` 相对于当前路径打开文件

`:find {filename}` 通过文件名打开文件，需要配置 `path` 选项，如`:set path+=app/**` 可以使得app 录下的文件更容易被找到

更快地移动及跳转

用动作命令在文档中移动

参考文档 `h motion.txt`，`:h word-motions`

`j` 向下移动一个实际行，`gj` 向下一个屏幕行，同理，`0` 跳转到行首，`^` 移动到第一个空白字符，`$` 移到行尾。

`w` 移动到下个单词的开头，`b` 反向移动到上一个单词的开头，`e` 正向移动到下一个单词的结尾。`ge` 向移动到上一个单词的结尾，`W`，`B`，`E`，`gE` 的移动单位是字串。

`f+` 当前行找到+并定位过去，`;` 查找下一个，`,` 上一个，`F` 当前行查找上一个，`t{char}` 移动到下一个{c art}所在之处的前一个字符上。`T` 反向移动到指定字符的后一个字符上。

`dt`. 删除当前光标到句尾所有文本，不包括句号。

`v/ge<CR>hd` 字符高亮，选中到ge单词处，向左移动光标，删除，等价于`d/ge<CR>`

`vi` 高亮选中{}内部的文本，不包括{}，类似还有`i>`，`i)`，`i]`，`i'`，`i"`，`it`，`t`表示tag，xml标签

`a"` 选中双引号括起来的文本，包括双引号，其他类似与上面i的用法

`c{motion}` 删除指定文本并切换到插入模式，如 `ci` 删除双引号内的文本并进入插入模式。类似还有 `dit`, `yit`

`iw`, `iW`, `is`, `ip` 当前单词，字串，句子，段落

`aw`, `aW`, `as`, `ap` 当前单词，字串，句子，段落及一个空格/空行

`%` 在匹配括号间跳转，可执行如下动作来修改匹配的括号(`%w{London Berlin New\ York}`): `dt{`, 除{之前的字符, `%`匹配到下一个括号, `r]`用]替换, ```` 跳回到上次位置, `r[` 替换为[

文件间跳转

`[count]G` 跳转到count指定行

`/pattern<CR> /?pattern<CR> /n /N` 跳转到下一个或上一个模式出现之处

`()`, 跳转到上一句或下一句的开头

`{}`, 跳转到上一段或下一段的开头

`H M L` 跳转到屏幕最上方、正中间、最下方

`<C->` 跳转到光标下的关键字的定义之处

`'{mark} ``{mark}`` 跳转到一个位置标记

寄存器

复制与粘贴

`xp` 替换光标之后的两个字符, `ddp`替换当前和下一行, `yyp` 复制当前行到下一行

`"ayiw` 讲当前单词复制到寄存器a中, `"bdd` 讲整行文本剪切至寄存器b中, `"ap` 粘贴来自寄存器a的单词, `"bp` 粘贴来自寄存器b的一整行文本。或者在命令模式下, 通过`:delete c` 剪切到寄存器c, 然后通过`:ut c`粘贴到光标所在行之下。

执行`y{motion}` 时, 讲复制到专用寄存器中, 可通过`"OP`粘贴

`"_d{motion}` 剪切至黑洞寄存器, 文本将不会覆盖任何寄存器的内容, 直接丢掉

`"+` 系统剪切板, 如果在外部用剪切或复制命令获取了文本, 可以通过`"+p` 获取, 或者在插入模式下用`<C-r>+`, 反之也可以用此讲vim文本中的内容复制到剪切板供外部程序使用

`"=` 表达式寄存器

`"%` 获取当前文件名

`"#` 获取轮换文件名

`".` 上次插入的文本

`:"` 上次执行的Ex命令

"/ 上次查找的模式

用寄存器中的内容替换高亮选区的文本：**yiw** 复制一个单词，**jww** 跳转到下一行的第二个单词，**ve** 亮选中，**p** 粘贴

交换两个单词：**fc** 定位到c开头的单词，**de** 剪切这个单词，**mm** 标记该处，**ww** 定位到下下一个单词，**e** 高亮选中，**p** 粘贴单词，**``m** 返回标记位，**P`** 粘贴。小写为粘贴到光标之后，大写为粘贴到光标之前