

Vim Notes

主要是 VIM 常用快捷键速查表，以及一份忘记从哪里抄来的配置文件。

- [Vim Notes](#)
 - [基础](#)
 - [光标移动与插入](#)
 - [查找与替换](#)
 - [删除、复制、粘贴](#)
 - [文件操作、多文件编辑](#)
 - [命令重做、宏](#)
 - [自动补全](#)
 - [拓展](#)
 - [代码块缩进](#)
 - [设置、系统命令、其他](#)
 - [VISUAL BLOCK 纵向编辑模式 & 查找匹配编辑模式](#)
 - [简明的 VIM 配置文件](#)
 - [其他](#)

基础

光标移动与插入

- `h, j, k, l` 单字符光标移动
- `i, a, o, O` 在当前字符之前、之后，当前行下一行、上一行插入
- `w, b, W, B` 单词光标移动，以空格为标准的单词光标移动
- `C+f, C+b, C+d, C+u` 上翻页，下翻页，上翻半页，下翻半页
- `gg, G, nG` 移动到文件第一行 / 最后一行 / 第 n 行
- `H, M, L` 移动到屏幕最上方，中间，最下方
- `(,), {, }, %` 前后移动一句，一段，括号匹配移动。

查找与替换

- `/word, ?word` 向下/上查找字符串
- `:n1,n2s/word1/word2/g` 在n1,n2行之间将word1替换为word2
- `:%s/word1/word2/gc` 全局将word1替换为word2并要求确认

删除、复制、粘贴

`x,X` del, backspace

`dd, ndd, d1G, dG, d$, d0` 删除整行, 删除 n 行, 删除从第一行到光标, 删除从光标到最后一行, 删除从光标到本行结尾 / 开始

`yy, nyy, y1G, yG, y0, y$` 复制

`P,p` 在光标上一行/下一行粘贴

从外部粘贴时, 可以 `:set paste` 避免粘贴时自动缩进引起的格式问题。之后 `:set nopaste` 取消该设置。

`J` 合并当前行和下一行。处理粘贴文字段时有用。

`daw, da}, da), da]` 删除当前单词, 删除当前括号内内容。

文件操作、多文件编辑

`:q, :q!, :wq, zz` 退出, 强制退出, 保存退出, 保存退出

`:w, :w!, :w filename, :n1,n2 w filename` 写入, 强制写入, 写入到文件, n1 – n2 行写入到文件

`:r filename` 在编辑的数据中读入另一个文件 (新文件内容插入到光标所在行之后)

`:e filename, :open filename` 打开文件

`:n, :N` 下一个文件、上一个文件

`:ls` 列出正在编辑的文件

`:sp :vsp` 竖直, 水平分屏

`C+w+hjkl` 窗口切换

`:tabe filename` : open file in new tab

`gt gT` : next or previous tab.

`number+gt` " go to the number-th tab.

命令重做、宏

`n+cmd` do cmd n times

`u` 复原前一个操作

`qq, @q; qa, @a` 录制宏到 q 寄存器, 调用 q 寄存器中的宏; 录制宏到 a 寄存器, 调用 a 寄存器中的宏

【保存宏】

在 `.vimrc` 中保存一行内容:

```
let @q = 'macro contents'
```

`macro contents` 可以通过键入 `"qp` 粘贴当前 `q` 寄存器内容。

自动补全

当前文件内部的关键字补全。在 Insert Mode 内， `Ctrl+N/Ctrl+P` 。

拓展

代码块缩进

`:10,100>` 第10行至第100行缩进

`:20,80<` 第20行至第80行反缩进

VISUAL 模式选择好需要缩进的行后， `>` 缩进一次， `6>` 缩进六次，按 `<` 反缩紧

设置、系统命令、其他

`:! ShellCMD` 运行shell 命令

`:set nu` 显示行号

`:set nonu` 不显示行号

`:noh` 退出高亮（用于搜索之后退出搜索高亮）

VISUAL BLOCK 纵向编辑模式 & 查找匹配编辑模式

这两项功能可以用来代替 Sublime 的多光标编辑模式。

VISUAL BLOCK 模式。按 `CTRL + v` 进入列选择模式。选中多列相似文本后，可以进行的操作包括：

- 按 `I` 进入行首插入模式，在行首插入文字。
- 按 `A` 在当前字符之后插入。
- 按 `r` 修改当前字符。

完成后改后按 `ESC` 退出并将修改应用到所有文本列。注意应用之后会彻底退出 VISUAL BLOCK 模式，而 Sublime 的多光标并不会自动消失。

查找匹配编辑模式。查找单词之后，按下 `cn` 对匹配的单词进行编辑（此时会自动删除匹配的单词，等待键入新的单词，目前没有办法保持这个单词进行编辑）。编辑完毕按 `ESC` 退出后，按下 `.` 将同样的操作应用到下一个 Match 或者按 `n` 跳过一个 Match。

复杂的操作还是建议使用宏完成。

简明的 VIM 配置文件

```
" 编码相关 =====
" 设置 VIM 内部编码格式
set encoding=utf-8
" 设置终端显示编码格式，通常与 Encoding 一致
set termencoding=utf-8
" 设置加载文件时启发式自动编码检测列表。
" 如果打开文件仍然乱码，可以调整一下该表顺序
set fileencodings=utf-8,gb2312,gbk,gb18030,big5,cp936,latin-1
" 写出文件编码格式
set fileencoding=utf-8
" 换行风格问题
set fileformat=unix

" 显示相关 =====
" 显示行号
set number
" 显示标尺
set ruler
" 打开语法高亮
syntax on

" 键入 : 之后开始键入命令时，按下 Tab 打开自动补全窗口
set wildmenu
" 命令行自动补齐顺序
set wildmode=longest:list;full
" 历史纪录
set history=1000
" 输入的命令显示出来，看的清楚些
set showcmd

" 高亮搜索的字符串
set hlsearch
" 搜索跳转时光标与顶部底部保持3行距离
set scrolloff=3
" Vertical bar in insert mode
let &t_SI = "\<Esc>]50;CursorShape=1\x7" "
" Block in normal mode
let &t_EI = "\<Esc>]50;CursorShape=0\x7" "

" 状态行显示的内容
set statusline=%F%m%r%h%w\ [FORMAT=%{&ff}]\ [TYPE=%Y]\ [POS=%l,%v][%p%%]\ %{strftime("\%d/%m/%y\ -\ %H:%M")}
" 总是显示状态栏
set laststatus=2
```

```
" 主题
colorscheme desert
" 指定配色方案是256色
set t_Co=256
" 去掉输入错误提示声音
set noeb

" 缩进相关 =====
" 去掉有关vi一致性模式，避免以前版本的一些bug和局限，解决backspace不能使用的问题
set nocompatible
set backspace=indent,eol,start
set backspace=2
" 启用自动对齐功能，把上一行的对齐格式应用到下一行
set autoindent
" 依据上面的格式，智能的选择对齐方式，对于类似C语言编写很有用处
set smartindent
" 用空格代替tab
set expandtab
" 设置显示制表符的空格字符个数,改进tab缩进值，默认为8，现改为4
set tabstop=4
" 统一缩进为4，方便在开启了et后使用退格(backspace)键，每次退格将删除X个空格
set softtabstop=4
" 设定自动缩进为4个字符，程序中自动缩进所使用的空白长度
set shiftwidth=4
" 设置帮助文件为中文(需要安装vimdoc文档)
set helplang=cn
" 显示匹配的括号
set showmatch
" C风格缩进
set cindent
set completeopt=longest,menu

" 自动重载保存备份等 =====
" vim禁用自动备份，阻止 VIM 创建乱七八糟的文件，但这样一来修改文件需慎重
" 默认不开启
" set nobackup
" set nowritebackup
" set noswapfile
" 失去焦点时自动保存，并忽略对于 untitled 文件的警告。注意该功能在命令行下不一定可靠
au FocusLost * silent! wa
" 切换 buff 等动作时自动保存
set autowriteall
" 共享剪贴板
set clipboard+=unnamed
```

```
" 文件被改动时自动载入
set autoread
```

```
" 检测文件的类型，并根据文件类型启动相应的插件和缩进方案（如果有）
filetype on
filetype plugin on
filetype indent on
```

其他

`ls | vim -` 从 `stdin` 读取文件。用于替代 `less` 等命令阅读较长的命令行输出。