# Vi Cmd

**dd** 删除整行

**nG** 跳到指定行

**yy** 拷贝一行

**p** 粘贴

**u** 取消上一次操作

**U** 取消所有操作

c**trl+r** 对撤销的撤销

**:e!** 放弃更改，然后相当于重新打开

**ma** 标记当前行为'a'

y`a 复制当前行至标记`a的位置

d`a 删除当前行至标记`a的位置

**%** 移动到匹配的括号上去，如{}, [], ()

**`** 移动到上一次的修改行

**v%** 将光标放到{处，执行v%就会选中{}中的内容

**fx** 向右跳转到字符x处

**Fx** 向左............

**tx** 向右跳转到字符x前

**Tx** 向左.............

**#** 到与当前单词相同的上一个单词

**\*** ..................下一个....

**:set hlsearch** 高亮显示查找结果

**:set nohlsearch** 取消高亮

**nG**：光标移至第n行首

**``**  回到原来行

**zz** 将当前行放置屏幕中间

**~** 更改字母的大小写

**.**  重复上一个命令

## 打开文件：

**vi filename :**打开或新建文件，并将光标置于第一行首

**vi +n filename** ：打开文件，并将光标置于第n行首

**vi + filename** ：打开文件，并将光标置于最后一行首

**vi +/pattern filename**：打开文件，并将光标置于第一个与pattern匹配的串处

## 光标移动：

**k、j、h、l**——上、下、左、右光标移动命令

**%**是从大括号的开始移动到大括号的结束位置

**:**后边加行号就是跳到这一行

**gg** 移动到文件第一行

**shift + g**移动到文件末尾

**)** ：光标移至句尾

**(** ：光标移至句首

**}**：光标移至段落开头

**{**：光标移至段落结尾

nG：光标移至第n行首

`` 回到原来行

n+：光标下移n行

n-：光标上移n行

n$：光标移至第n行尾

0：（注意是数字零）光标移至当前行首

^:移动光标到行首；

$:移动光标到行尾；

shift -移动到光标所在行行首

shift +移动到光标所在行下一行首

H ：光标移至屏幕顶行

M ：光标移至屏幕中间行

L ：光标移至屏幕最后行

ctrl-b:类似于键盘上的"PgUp"(b为backword)

ctrl-f：类似于键盘上的"PgDn"(f为forword)

w:光标移动到下一个单词的词首；注：对于中文，连续的多个汉字作为一个word。

2w:重复执行w操作2次；

e:光标移动到下一个单词的词尾；

5e:重复执行e操作5次；

b：向前移动光标，移动到前一个单词的词首；

m+单词（单词为a-z或者A-Z）

ma, 设定当前行标记为'a', 输入`a, 则会返回到标记为a的地方

mb, 设定当前行标记为'b', 输入`b, 则会返回到标记为b的地方

:delmarks 删除指定标记

:marks 列出所有的标记

' 跳转前的光标位置

" 最后编辑的光标位置

[ 最后修改的开始位置

] 最后修改的结束位置

## 删除命令：

ndw/ndW：删除光标处开始及其后的n-1个字

do：删至行首

d$：删至行尾

ndd：删除当前行及其后n-1行

x/X：删除一个字符，x删除光标后的，而X删除光标前的

**J**:删除换行符，使下一行并过来

**nJ:** 连接后面的n行

## 搜索及替换命令

:[range]s/{pattern}/{string}/[flag]

[range]通常是两个整数，表示作用域起始行和结束行的行号，中间以一个逗号隔开。如："4,8"表示第4行到第8行（包括第4行和第8行）。

百分号'%'表示文档的所有行。

省略不写则代表光标所在行，

"$"代表文本的最后一行

{pattern}和{string}都支持正则表达式。

{pattern}为空串，则使用上一条替换命令中的 {pattern}作为模式串pattern

{pattern}和{string}中不能包含分隔符，否则会产生歧义。为此，要使用反斜杠'\'进行转义。

如:"%s/<br>/<br />/g"必须写成":%s/<br>/<br \/>/g"

如果这两个表达式中的斜杠非常多，比如"file:///usr/share/man/man1/vim.1.gz"，那么可以使用其他字符作为分隔符（总是将作为替换命令代表的's'后的第一个字符作为分隔符）

如:%s+<br>+<br />+g ('+'加号代表分隔符)

[flag]有多种，"g"表示替换[range]范围内每行中的所有匹配模式，而不写"g"时，只对每行第一次匹配出现的{pattern}进行替换。

"c"表示每次替换前要求确认，

"i"表示不区分大小写，

/pattern：从光标开始处向文件尾搜索pattern

?pattern：从光标开始处向文件首搜索pattern

n：在同一方向重复上一次搜索命令

N：在反方向上重复上一次搜索命令

:s/str1/str2/ 用字符串 str2 替换行中首次出现的字符串 str1

:s/str1/str2/g 用字符串 str2 替换行中所有出现的字符串 str1

:.,$ s/str1/str2/g 用字符串 str2 替换正文当前行到末尾所有出现的字符串 str1

:1,$ s/str1/str2/g 用字符串 str2 替换正文中所有出现的字符串 str1

:g/str1/s//str2/g 功能同上

g 放在命令末尾，表示对搜索字符串的每次出现进行替换；

不加 g，表示只对搜索字符串的首次出现进行替换；

g 放在命令开头，表示对正文中所有包含搜索字符串的行进行替换操作。

## 搜索关键字高亮：支持记录3种搜索高亮

：match Search /main/ //搜索关键字“main”, 并按Search定义的字体颜色显示, 字体颜色详见hi命令

: 2macth Search /main1/

: 3match Search /main2/

## 同时显示多个文件：

:split

:vsplit

## 1.文件间切换

Ctrl+6—下一个文件

:bn—下一个文件

:bp—上一个文件

对于用(v)split在多个窗格中打开的文件，这种方法只会在当前窗格中切换不同的文件。

## 2.在窗格间切换的方法

Ctrl+w+方向键——切换到前／下／上／后一个窗格

Ctrl+w+h/j/k/l ——同上

Ctrl+ww——依次向后切换到下一个窗格中

## 打开指定函数：

vim -t main

:ts //如果有多个main的话，执行这个命令会列出所有main函数入口

## 复制指定字符：

在命令行下按"v"进入可视模式，然后按"H"或"L"移动光标，选中要复制的字符，

然后按一次"y"。。。这样就复制下来了，如果要剪切就按“d“

## vim命令：

set ignorecase

set noignorecase

## 折叠代码

zf进行折叠,

zo打开折叠,也可以方向键向右打开折叠,

zc 关闭折叠(只要在被折叠的块中的任一个语句就行)

缩进代码

# Ctags

git clone <https://github.com/SublimeText/CTags>

or

sudo apt-get install exuberant-ctags

sudo apt-get install ctags

$ cd /home/aepc/clxu/vlc-2.1.4

$ ctags -R //在/home/aepc/clxu/vlc-2.1.4目录下会生成一个 tags 文件, -R表示递归子文件夹

$ vim /home/aepc/clxu/vlc-2.1.4/bin/vlc.c

:set tags=/home/aepc/clxu/vlc-2.1.4/tags //该命令将tags文件加入到vim中来

ctrl+], :tselect 列出光标所在函数的所有定义处

Ctrl+] 直接跳转到第一个 func tag 处

g] 不跳转，列出有多少个 func tags 可选

g Ctrl+] 一般同 g]，但如果 func tag 只有一个，则直接跳转过去

ctrl+T, 这样又跳回到函数被调用的地方

ctrl+N, 自动补全

# cscope

sudo apt-get install cscope

$ cd /home/aepc/clxu/vlc-2.1.4

$ cscope -Rbkq //在/home/aepc/clxu/vlc-2.1.4目录下会生成cscope.out, cscope.in.out, cscope.po.out

//cscope.out是基本的符号索引，后两个文件是使用”-q”选项生成的，可以加快cscope的索引速度。

-R: 在生成索引文件时，搜索子目录树中的代码

-b: 只生成索引文件，不进入cscope的界面

-k: 在生成索引文件时，不搜索/usr/include目录

-q: 生成cscope.in.out和cscope.po.out文件，加快cscope的索引速度

//cscope在产生索引文件时，只搜索类型为C, lex和yacc的文件(后缀名为.c, .h, .l, .y)，C++的文件根本没有生成索引。不过按照手册上的说明，cscope支持c++和Java语言的文件。

c++:

find . -name "\*.h" -o -name "\*.c"-o -name "\*.cc" > cscope.files

cscope -bkq -i cscope.files

ctags -R

:cs add ./cscope.out

:cs find c soap\_init1 //find caller

:cs find c soap\_init1 //find definition

# Taglist

1. 从http://www.vim.org/scripts/script.php?script\_id=273下载安装包.

2. 进入~/.vim目录，将Taglist安装包解压，解压后会在~/.vim目录中生成几个新子目录，如plugin和doc（安装其它插件时，可能还会新建autoload等其它目录）。

3. 进入~/.vim/doc目录，在Vim下运行"helptags ."命令。此步骤是将doc下的帮助文档加入到Vim的帮助主题中，这样我们就可以通过在Vim中运行“help taglist.txt”查看taglist帮助。

4. 打开配置文件~/.vimrc，加入以下几行：

let Tlist\_Show\_One\_File=1 "不同时显示多个文件的tag，只显示当前文件的

let Tlist\_Exit\_OnlyWindow=1 "如果taglist窗口是最后一个窗口，则退出vim

let Tlist\_Ctags\_Cmd="/usr/bin/ctags" "将taglist与ctags关联

基本功能使用方法

在Vim命令行下运行":Tlist"就可以打开Taglist窗口，再次运行":Tlist"则关闭。

左右窗口切换Ctrl+ww

在taglist窗口中，可以使用下面的快捷键：

<CR> 跳到光标下tag所定义的位置，用鼠标双击此tag功能也一样

o 在一个新打开的窗口中显示光标下tag

<Space> 显示光标下tag的原型定义

u 更新taglist窗口中的tag

s 更改排序方式，在按名字排序和按出现顺序排序间切换

x taglist窗口放大和缩小，方便查看较长的tag

+ 打开一个折叠，同zo

- 将tag折叠起来，同zc

\* 打开所有的折叠，同zR

= 将所有tag折叠起来，同zM

[[ 跳到前一个文件

]] 跳到后一个文件

q 关闭taglist窗口

<F1> 显示帮助

# End