常用操作

> 进入一般模式为开始编辑，或者按esc后进入

按键	效果
a,i,r,o,A,I,R,O	进入编辑模式
h,backspace	左移动
l,space	右移动
j	下移动
k	上移动
0,	移动到行首
$	移动到行末， `1$` 表示当前行的行尾， `2$` 表示当前行的下一行的行尾
b	按照单词向前移动 字首
e	按照单词向后移动 字尾
w	按照单词向后移至次一个字首
H	移动到屏幕最上 非空白字
M	移动到屏幕中央 非空白字
L	移动到屏幕最下 非空白字
G	移动到文档最后一行
gg	移动到文档第一行
v	进入光标模式，配合移动键选中多行
Ctrl+f	向下翻页
Ctrl+b	向上翻页
u	撤销上一次操作
``	回到上次编辑的位置
dw	删除这个单词后面的内容
dd	删除光标当前行

按键	效果
dG	删除光标后的全部文字
d$	删除本行光标后面的内容
d0	删除本行光标前面的内容
y	复制当前行，会复制换行符
yy	复制当前行的内容
yyp	复制当前行到下一行，此复制不会放到剪切板中
nyy	复制当前开始的n行
p,P,.	粘贴
ddp	当前行和下一行互换位置
J	合并行
Ctrl+r	重复上一次动作
Ctrl+z	暂停并退出
ZZ	保存离开
xp	交换字符后面的交换到前面
~	更换当前光标位置的大小写，并光标移动到本行右一个位置，直到无法移动

光标详细操作

按键	效果
Ctrl+e	向下滚动
Ctrl+b	向上翻页
b	按照单词向前移动 字首
B	按照单词向前移动 字首 忽略一些标点符号
e	按照单词向后移动 字尾
E	按照单词向后移动 忽略一些标点符号
w	按照单词向后移至次一个字首
W	按照单词向后移至次一个字首 忽略一些标点符号

按键	效果
H	移动到屏幕最上 非空白字
M	移动到屏幕中央 非空白字
L	移动到屏幕最下 非空白字
G	移动到文档最后一行
gg	移动到文档第一行
(	光标到句尾
)	光标到局首
{	光标到段落开头
}	光标到段落结尾
nG	光标下移动到n行的首位
n$	光标移动到n行尾部
n+	光标下移动n行
n-	光标上移动n行

查找命令

指令	效果
*	向下查找同样光标的字符
#	向上查找同样光标的字符
/code	查找 code 一样的内容，向后
?code	查找 code 一样的内容，向前
n	查找下一处
N	查找上一处
ma	在光标处做一个名叫a的标记 可用26个标记 (a~z)
`a	移动到一个标记a
d`a	删除当前位置到标记a之间的内容
:marks	查看所有标记

指令模式

指令	效果
:q	一般退出
:q!	退出不保存
:wq	保存退出
:w filename	另存为 filename
:jumps	历史编辑文档记录
:set nu	设置行号显示
:set nonu	取消行号显示
:set	显示设置参数
:set autoindent	自动缩排，回车与第一个非空格符对齐
:syntax on/off	根据程序语法高亮显示
:set highlight	高亮设置查看
:set hlsearch	查找代码高亮显示
:nohlsearch	暂时关闭高亮显示
:set nohlsearch	永久关闭高亮显示
:set bg=dark	设置暗色调
:set bg=light	设置亮色调

屏幕翻滚

按键	效果
Ctrl+f	向文件尾翻一屏幕
Ctrl+b	向文件首翻一屏幕
Ctrl+d	向文件尾翻半屏幕
Ctrl+u	向文件首翻半屏幕

插入命令

按键	效果
i	在光标前
I	在当前行首
a	在光标后
A	在当前行尾部
o	在当前行下新开一行
O	在当前行上新开一行
r	替换当前字符
R	替换当前行及后面的字符，直到按esc为止
s	从当前行开始，以输入的文本替代指定数目的字符
S	删除指定数目的行，并以输入的文本替代
ncw,nCW	修改指定数目n的字符
nCC	修改指定数目n的行

删除命令

按键	效果
ndw,nDW	删除光标开始及其后 n-1 个字符
dw	删除这个单词后面的内容
dd	删除光标当前行
dG	删除光标后的全部文字
d$	删除本行光标后面的内容
d0	删除本行光标前面的内容
ndd	删除当前行，以及其后的n-1行
x	删除一个字符，光标后
X	删除一个字符，光标前
Ctrl+u	删除输入模式下的输入的文本

多窗口模式

指令	效果
:split	创建新窗口
Ctrl+w	切换窗口
Ctrl-w =	所有窗口一样高
Ctrl-w+方向键	多窗口视图切换

多文件编辑

指令	效果	
:args	列出当前编辑的文件名	
:next	打开多文件，使用 n(Next) p(revious)	N(ext) 切换
:file	列出当前打开的所有文件	

vim 自定义技巧

复制粘贴取消缩进

```
:set paste
```

进入paste模式以后，可以在插入模式下粘贴内容，不会有任何变形
这个参数做了这么多事：

```
textwidth设置为0
wrapmargin设置为0
set noai
set nosi
softtabstop设置为0
revins重置
ruler重置
showmatch重置
formatoptions使用空值
```

下面的选项值不变，但却被禁用

```
lisp
indentexpr
cindent
```

绑定快捷键来激活/取消 paste模式

```
:set pastetoggle=<F11>
```

出现粘贴换行符错位，设置一下 `.vimrc`

```
" this can change way of paste words
:set paste
" default tabstop=8
:set tabstop=4
" use keyboard F11 to  change paste mode
:set pastetoggle=<F11>
```

vim 缩进

Normal Mode下，命令 `>>` 将对当前行增加缩进，而命令 `<<` 则将对当前行减少缩进
在命令前使用数字，来指定命令作用的范围

```
5<<
```

在Insert/Replace Mode下

`Ctrl-Shift-t` 可以增加当前行的缩进
`Ctrl-Shift-d` 则可以减少当前行的缩进
使用 `0-Ctrl-Shift-d` 命令，将移除所有缩进

> 需要注意的是，当我们输入命令中的“0”时，Vim会认为我们要在文本中插入一个0，并在屏幕上显示输入的“0”；然后当我们执行命令0-Ctrl-Shift-d时，Vim就会意识到我们要做的是减少缩进，这时0会就会从屏幕上消失

vim tab缩进

tab缩进宽度默认为8个空格

我们可以使用以下命令，来修改缩进宽度

```
:set tabstop=4
:set softtabstop=4
:set shiftwidth=4
:set expandtab
```

- tabstop:表示一个 tab 显示出来是多少个空格的长度默认 8

- softtabstop:表示在编辑模式的时候按退格键的时候退回缩进的长度当使用 expandtab 时特别有用。

- shiftwidth:表示每一级缩进的长度□一般设置成跟 softtabstop 一样。 当设置成 expandtab 时□缩进用空格来表示□noexpandtab 则是用制表符表示一个缩进

- expandtab选项，用来控制是否将Tab转换为空格,但是这个选项并不会改变已经存在的文本，如果需要应用此设置将所有Tab转换为空格，需要执行

```
:retab!
```

vim 自动缩进

- cindent

```
:set cindent
```

vim可以很好的识别出C和Java等结构化程序设计语言，并且能用C语言的缩进格式来处理程序的缩进结构

- smartindent

```
:set smartindent
```

在这种缩进模式中，每一行都和前一行有相同的缩进量，同时这种缩进形式能正确的识别出花括号，当遇到右花括号（}），则取消缩进形式。此外还增加了识别C语言关键字的功能。如果一行是以#开头的，那么这种格式将会被特殊对待而不采用缩进格式。

- autoindent

```
:set autoindent
```

在这种缩进形式中，新增加的行和前一行使用相同的缩进形式

显示隐藏符号

- 默认不显示 `:set nolist`
- 显示 `:set invlist`

```
" normal is :set nolist | show hide is :set invlist
:set nolist
```

使用vim寄存器

使用vim寄存器 “+p 粘贴

根本不用考虑是否自动缩进，是否paste模式，直接原文传递

如果想保存原寄存器中内容而同时增加新的内容
就要在yy前增加标签
标签以双引号开始，跟着的是标签名称，可以是数字0-9，也可以是26个字母

显示所有寄存器内容

```
:reg
```

注意两个特殊的寄存器： `*` 和 `+`

这两个寄存器是和系统相通的，前者关联系统选择缓冲区，后者关联系统剪切板
通过它们可以和其他程序进行数据交换

> 若寄存器列表里无 `*` 或 `+` 寄存器，则可能是由于没有安装vim的图形界面所致
> sudo apt-get install vim-gnome

设置vim永远显示行号

修改vim的配置文件加入 set nu

```
vi ~/.vimrc
```

然后输入

```
set nu
```

当然也可以输入其他配置类似

```
set nonu
syntax on
```

vimrc 常用配置

```
" open syntax
syntax on
" set not show line number can change by :set nu
:set nonu
" set show line number when in edit
:set ruler
" set tab button stop
" default tabstop=8
:set tabstop=4
:set softtabstop=4
:set shiftwidth=4
:set expandtab
" use keyboard F11 to change paste mode
:set pastetoggle=<F11>
" normal is :set nolist | show hide is :set invlist
:set nolist

" fix mac vim keyboard delete can not delete error, so as set backspace=indent,eol,start
set backspace=2
" ------------
" install plug-in manager see https://github.com/VundleVim/Vundle.vim
```

查看vim设置的样例文件

- linux 查看

```
find /usr/share/ -name "*example.vim"
```

- mac 查看方法

```
locate example.vim
```

> 如果是第一次运行会报告错误，需要建立索引，根据提示操作即可，建议运行一次 updatedb

找到标识为 `example.vim` 的文件就是样例