以:和/开头的命令都有历史纪录，可以首先键入:或/然后按上下箭头来选择某个历史命令。

**启动vim**

在命令行窗口中输入以下命令即可

vim 直接启动vim

vim filename 打开vim并创建名为filename的文件

**文件命令**

打开单个文件

vim file

同时打开多个文件

vim file1 file2 file3 ...

在vim窗口中打开一个新文件

:open file

在新窗口中打开文件

:split file

切换到下一个文件

:bn

切换到上一个文件

:bp

查看当前打开的文件列表，当前正在编辑的文件会用[]括起来。

:args

打开远程文件，比如ftp或者share folder

:e ftp://192.168.10.76/abc.txt

:e \\qadrive\test\1.txt

**vim的模式**

正常模式（按Esc或Ctrl+[进入） 左下角显示文件名或为空  
插入模式（按i键进入） 左下角显示--INSERT--  
可视模式（不知道如何进入） 左下角显示--VISUAL--

**导航命令**

% 括号匹配

**插入命令**

i 在当前位置生前插入

I 在当前行首插入

a 在当前位置后插入

A 在当前行尾插入

o 在当前行之后插入一行

O 在当前行之前插入一行

**查找命令**

/text　　查找text，按n健查找下一个，按N健查找前一个。

?text　　查找text，反向查找，按n健查找下一个，按N健查找前一个。

vim中有一些特殊字符在查找时需要转义　　.\*[]^%/?~$

:set ignorecase　　忽略大小写的查找

:set noignorecase　　不忽略大小写的查找

查找很长的词，如果一个词很长，键入麻烦，可以将光标移动到该词上，按\*或#键即可以该单词进行搜索，相当于/搜索。而#命令相当于?搜索。

:set hlsearch　　高亮搜索结果，所有结果都高亮显示，而不是只显示一个匹配。

:set nohlsearch　　关闭高亮搜索显示

:nohlsearch　　关闭当前的高亮显示，如果再次搜索或者按下n或N键，则会再次高亮。

:set incsearch　　逐步搜索模式，对当前键入的字符进行搜索而不必等待键入完成。

:set wrapscan　　重新搜索，在搜索到文件头或尾时，返回继续搜索，默认开启。

**替换命令**

ra 将当前字符替换为a，当期字符即光标所在字符。

s/old/new/ 用old替换new，替换当前行的第一个匹配

s/old/new/g 用old替换new，替换当前行的所有匹配

%s/old/new/ 用old替换new，替换所有行的第一个匹配

%s/old/new/g 用old替换new，替换整个文件的所有匹配

:10,20 s/^/    /g 在第10行知第20行每行前面加四个空格，用于缩进。

ddp 交换光标所在行和其下紧邻的一行。

**移动命令**

h 左移一个字符  
l 右移一个字符，这个命令很少用，一般用w代替。  
k 上移一个字符  
j 下移一个字符  
以上四个命令可以配合数字使用，比如20j就是向下移动20行，5h就是向左移动5个字符，在Vim中，很多命令都可以配合数字使用，比如删除10个字符10x，在当前位置后插入3个！，3a！<Esc>，这里的Esc是必须的，否则命令不生效。

w 向前移动一个单词（光标停在单词首部），如果已到行尾，则转至下一行行首。此命令快，可以代替l命令。

b 向后移动一个单词 2b 向后移动2个单词

e，同w，只不过是光标停在单词尾部

ge，同b，光标停在单词尾部。

^ 移动到本行第一个非空白字符上。

0（数字0）移动到本行第一个字符上，

<HOME> 移动到本行第一个字符。同0健。

$ 移动到行尾 3$ 移动到下面3行的行尾

gg 移动到文件头。 = [[

G（shift + g） 移动到文件尾。 = ]]

f（find）命令也可以用于移动，fx将找到光标后第一个为x的字符，3fd将找到第三个为d的字符。

F 同f，反向查找。

跳到指定行，冒号+行号，回车，比如跳到240行就是 :240回车。另一个方法是行号+G，比如230G跳到230行。

Ctrl + e 向下滚动一行

Ctrl + y 向上滚动一行

Ctrl + d 向下滚动半屏

Ctrl + u 向上滚动半屏

Ctrl + f 向下滚动一屏

Ctrl + b 向上滚动一屏

**撤销和重做**

u 撤销（Undo）  
U 撤销对整行的操作  
Ctrl + r 重做（Redo），即撤销的撤销。

**删除命令**

x 删除当前字符

3x 删除当前光标开始向后三个字符

X 删除当前字符的前一个字符。X=dh

dl 删除当前字符， dl=x

dh 删除前一个字符

dd 删除当前行

dj 删除上一行

dk 删除下一行

10d 删除当前行开始的10行。

D 删除当前字符至行尾。D=d$

d$ 删除当前字符之后的所有字符（本行）

kdgg 删除当前行之前所有行（不包括当前行）

jdG（jd shift + g）   删除当前行之后所有行（不包括当前行）

:1,10d 删除1-10行

:11,$d 删除11行及以后所有的行

:1,$d 删除所有行

J(shift + j)　　删除两行之间的空行，实际上是合并两行。

**拷贝和粘贴**

yy 拷贝当前行

nyy 拷贝当前后开始的n行，比如2yy拷贝当前行及其下一行。

p  在当前光标后粘贴,如果之前使用了yy命令来复制一行，那么就在当前行的下一行粘贴。

shift+p 在当前行前粘贴

:1,10 co 20 将1-10行插入到第20行之后。

:1,$ co $ 将整个文件复制一份并添加到文件尾部。

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按y即可复制

ddp交换当前行和其下一行

xp交换当前字符和其后一个字符

**剪切命令**

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按d即可剪切

ndd 剪切当前行之后的n行。利用p命令可以对剪切的内容进行粘贴

:1,10d 将1-10行剪切。利用p命令可将剪切后的内容进行粘贴。

:1, 10 m 20 将第1-10行移动到第20行之后。

**退出命令**

:wq 保存并退出

ZZ 保存并退出

:q! 强制退出并忽略所有更改

:e! 放弃所有修改，并打开原来文件。

**窗口命令**

:split或new 打开一个新窗口，光标停在顶层的窗口上

:split file或:new file 用新窗口打开文件

split打开的窗口都是横向的，使用vsplit可以纵向打开窗口。

Ctrl+ww 移动到下一个窗口

Ctrl+wj 移动到下方的窗口

Ctrl+wk 移动到上方的窗口

关闭窗口

:close 最后一个窗口不能使用此命令，可以防止意外退出vim。

:q 如果是最后一个被关闭的窗口，那么将退出vim。

ZZ 保存并退出。

关闭所有窗口，只保留当前窗口

:only

录制宏

按q键加任意字母开始录制，再按q键结束录制（这意味着vim中的宏不可嵌套），使用的时候@加宏名，比如qa。。。q录制名为a的宏，@a使用这个宏。

**执行shell命令**

:!command

:!ls 列出当前目录下文件

:!perl -c script.pl 检查perl脚本语法，可以不用退出vim，非常方便。

:!perl script.pl 执行perl脚本，可以不用退出vim，非常方便。

:suspend或Ctrl - Z 挂起vim，回到shell，按fg可以返回vim。

**注释命令**

perl程序中#开始的行为注释，所以要注释某些行，只需在行首加入#

3,5 s/^/#/g 注释第3-5行

3,5 s/^#//g 解除3-5行的注释

1,$ s/^/#/g 注释整个文档。

:%s/^/#/g 注释整个文档，此法更快。

**帮助命令**

:help or F1 显示整个帮助  
:help xxx 显示xxx的帮助，比如 :help i, :help CTRL-[（即Ctrl+[的帮助）。  
:help 'number' Vim选项的帮助用单引号括起  
:help <Esc> 特殊键的帮助用<>扩起  
:help -t Vim启动参数的帮助用-  
：help i\_<Esc> 插入模式下Esc的帮助，某个模式下的帮助用模式\_主题的模式  
帮助文件中位于||之间的内容是超链接，可以用Ctrl+]进入链接，Ctrl+o（Ctrl + t）返回

**其他非编辑命令**

. 重复前一次命令

:set ruler?　　查看是否设置了ruler，在.vimrc中，使用set命令设制的选项都可以通过这个命令查看

:scriptnames　　查看vim脚本文件的位置，比如.vimrc文件，语法文件及plugin等。

:set list 显示非打印字符，如tab，空格，行尾等。如果tab无法显示，请确定用set lcs=tab:>-命令设置了.vimrc文件，并确保你的文件中的确有tab，如果开启了expendtab，那么tab将被扩展为空格。

Vim教程  
在Unix系统上  
$ vimtutor  
在Windows系统上  
:help tutor  
  
:syntax 列出已经定义的语法项  
:syntax clear 清除已定义的语法规则  
:syntax case match 大小写敏感，int和Int将视为不同的语法元素  
:syntax case ignore 大小写无关，int和Int将视为相同的语法元素，并使用同样的配色方案

## 取消对单词高亮

:noh